Application Software for Industrial Vision Systems

NeuroCheck
User Manual
Version 6.0

For Microsoft® Windows® 7 / Vista® / XP

Trademarks and Imprint

NeuroCheck GmbH
Neckarstraße 76/1
D-71686 Remseck
Germany
Phone: +49 (0) 7146 8956-0
Fax: +49 (0) 7146 8956-29

www.neurocheck.com

Version 6.0 of May 27, 2011

Table Of Contents

Welcome

Welcome to NeuroCheck, the leading platform for machine vision on Microsoft Windows. With NeuroCheck you have chosen a professional software product that has stood the test in thousands of industrial vision applications.

What's new in NeuroCheck 6.0?

With NeuroCheck 6.0 we're presenting our latest product version providing new functionality as well as optimizing and extending existing and proven functions. More than 2,000 suggestions from our customers and business partners were collected systematically and implemented in the new software version by our development team.

In the following you'll find an overview of the new functionality and improvements in NeuroCheck 6.0:

- NeuroCheck 6.0 uses the Microsoft .NET framework and supports the 64-bit edition of the Windows 7 operating system.
- The software's multi-threading architecture and many image processing algorithms utilize the computing power of modern multi core CPUs.
- The user interface of the manual mode of operation has been optimized according to ergonomic criteria providing new wizards and improved dialogs.
- The user interface and visualization of the automated inspection process can be adjusted to specific needs and designed freely using integrated graphic design tools.
- 25 new check functions have been added to the software. More than 20 check functions have been extended and improved. All image processing check functions now fully support color images.
- The communication interface for data and signal exchange has been significantly extended and provides quick and flexible integration into networked production environments.

The User Manual

This manual describes all program functions of NeuroCheck 6.0 in detail and provides a comprehensive reference documentation for the NeuroCheck software. The manual is not only intended as a reference work but also provides detailed understanding of the program concepts.

Because of the extensive functionality of the NeuroCheck software, the printed manual is limited to frequently used program functions. A detailed description of all program functions can be found in the online version of the manual that you can access by clicking on the menu item ? ▶ Help Topics ▶ User Manual.

Content

This User Manual is organized as follows:

- Chapter 1 („Basics“) explains the basic concepts and the most important terms of the NeuroCheck software.
- Chapter 2 („Check Routine“) describes how to create and configure check routines as the central element of the visual inspection application.
- Chapter 3 („Hardware Integration“) tells you how to integrate cameras and communication devices into the system.
- Chapter 4 („Automatic Mode“) depicts how to configure the automatic mode in advance, and how to operate it during run-time.
- Chapter 5 („Personalize Software“) shows you how to adapt NeuroCheck to the individual needs of your application.
- Chapter 6 („Special Applications“) deals with advanced and special subjects.
- Chapter 7 („Appendix“) is for reference purposes, e.g., a glossary.

Additional help resources

Getting Started:

This introduction to NeuroCheck will help you to install the system, will show you how to create your first image processing application using NeuroCheck, and how to get further information about how to work with NeuroCheck. The Getting Started manual is the ideal starting point for your entry into working with the NeuroCheck software when creating your first visual inspection application interactively.

Training Course:

This manual will acquaint you with the operation of the NeuroCheck software and present you the many fields of application in detail. It will guide you step by step through configuring automatic visual inspection applications. The Training Course manual is suitable for self study but also as a supporting document in a seminar.

Operator's manual (in preparation):

This manual will introduce you to NeuroCheck and show you how to gain comprehensive information about working with NeuroCheck; it will help you to start up a visual inspection system and support you if errors occur. It focuses on process optimization, error diagnostics, data safety, maintenance and support. The Operator's manual is intended for commissioning engineers and operators of NeuroCheck visual inspection systems.

Online help:

The online help system provides you with the information you need and where you need it when working with the NeuroCheck software, whether in the lab or on the line. Using the NeuroCheck user interface in manual mode, open the help system by clicking on the menu item ? ▶ Help Topics. In addition you can access the appropriate information in the help system from most program dialogs.

NeuroCheck Software Support:

E-mail: support@neurocheck.com
Phone: +49 (0)711-229646-31

We wish you fun reading the manual and success with your work with the NeuroCheck software!

1. Basics

This chapter gives you an overview of the basic concepts of the NeuroCheck Software and explains the most important terms.

> If you are using NeuroCheck for the first time, we recommend you read the **Getting Started Manual** first. It contains easy-to-read introductory chapters as well as a step by step guide to creating an inspection solution.
>
> Additionally, we would like to inform you about our **training courses**. For details and dates please refer to our web site http://www.neurocheck.com

1.1 What is NeuroCheck?

NeuroCheck is a universal machine vision software for all areas of automatic visual inspection in industrial manufacturing. NeuroCheck offers an integrated environment for the interactive configuration of visual inspection applications and their fully automated execution on the manufacturing line.

With its broad selection of image processing functions, NeuroCheck can be used in a wide range of application areas. Highly automated check routine configuration and an intuitive graphical user interface help to solve visual inspection problems. NeuroCheck reduces cost by easy set-up, reconfiguration and rapid turn-around.

The following simplified figure shows the basic setup of a visual inspection system with the required hardware components and communication connections:

The main features of NeuroCheck are:

Machine Vision Functions

The software encapsulates thousands of powerful and long-proven machine vision functions of the NeuroCheck image processing library. The check functions are logically grouped into categories such as image capturing, image pre-processing, machine vision analysis and measuring.

Camera integration

NeuroCheck supports a large number of modern digital cameras according to FireWire™ a/b and Gigabit Ethernet standards with various resolutions. The unique NeuroCheck device manager makes setup and configuration of cameras simple and easy.

Manual Mode

Manual mode presents a development interface for creating an inspection solution with an interactive graphical approach. Here you can determine the logical structure of the check procedure and the parameters for executing the inspection process.

Automatic Mode

In automatic mode, the previously configured check runs automatically. The system monitor visualizes status and results in freely configurable windows. Process control is effected via PLC or master computer or manual intervention by an operator.

Process integration

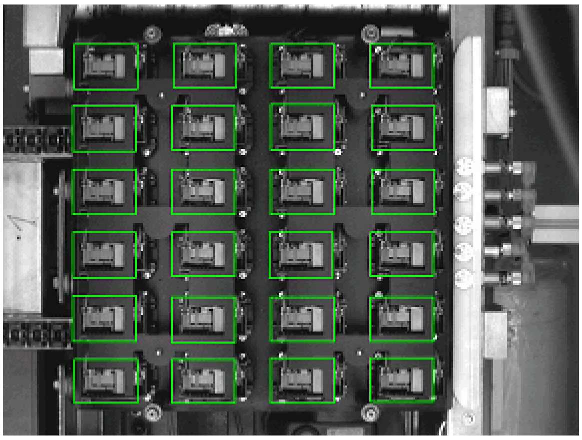
For communicating with a supervisory control system, a number of modern standard interfaces are available. Execution of the inspection procedure can be affected dynamically by the process peripherals. NeuroCheck also sends the inspection results and measurement values to the peripherals.

1.2 Application Areas

This section gives a brief introduction to typical visual inspection tasks NeuroCheck can be used for. You will find examples for all these application areas on the NeuroCheck Web site http://www.neurocheck.com and in the reference book **Industrial Image Processing** (1999, ISBN 3-540-66410-6) published by Springer Verlag (Berlin Heidelberg New York), which presents solutions from all areas of industrial visual inspection created with NeuroCheck.

1.2.1 Presence Verification / Completeness Check

Use NeuroCheck to detect the presence of all required parts and components easily and reliably.

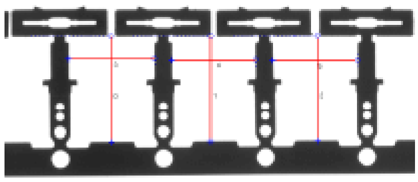


Examples for this type of application are:

- Check presence of connector pins.
- Check number of tablets in a blister pack or the completeness of packaging contents in general.
- Check circuit board components.
- Check correct injection mold of plastic parts.
- Inspect work pieces for damage (using appropriate measurements).
- Check presence of screws, contacts, markings, inscriptions etc.
- Check fill level and correct closure of bottles.

1.2.2 Gauging / Dimensional Inspection

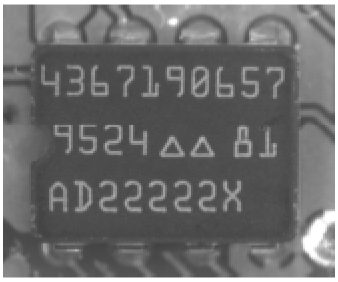
You can use NeuroCheck to check every aspect of the geometry of a work piece, e.g. distances between objects, angles between edges, parallelism, concentricity… Special subpixel algorithms allow for measuring with an accuracy of up to a fraction of an image pixel.

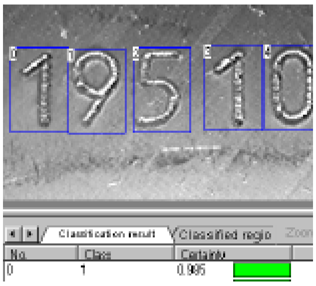


1.2.3 Character Recognition (OCR, OCV)

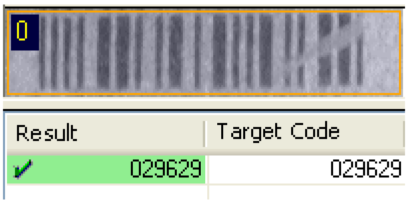
NeuroCheck uses adaptive classifiers for character recognition. They can be trained to recognize characters in fonts created by marking technology used in industry, for example laser-engraving, stamping, printing…

Combined with NeuroCheck's versatile image processing functionality, this technology can be used for a wide variety of recognition tasks, for example company logos or special codes.

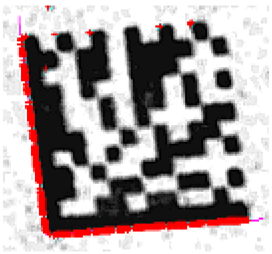




1.2.4 Bar Code and DataMatrix Code Identification



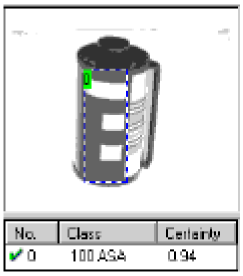
Examples of bar code identification can be found in all industrial areas, as different types of bar codes are used for the identification of all kinds of products in packaging, logistics, manufacturing control... NeuroCheck reads all common bar codes with contrast levels as low as 5%, forward and backward as well as in horizontal or vertical direction and even rotated.



The two-dimensional DataMatrix code offers higher storage capacity than conventional bar codes and increased robustness through the advanced ECC-200 method for error correction. NeuroCheck's highly configurable DataMatrix identification function detects and decodes DataMatrix codes at any angle of rotation, in negative or positive print and under difficult surface conditions.

1.2.5 Pattern Recognition

Parts can also be identified by readable letters or arbitrary symbols instead of bar codes. The neural networks used by NeuroCheck can learn from examples to recognize arbitrary patterns, enabling you to fine tune the recognition process to your particular application.

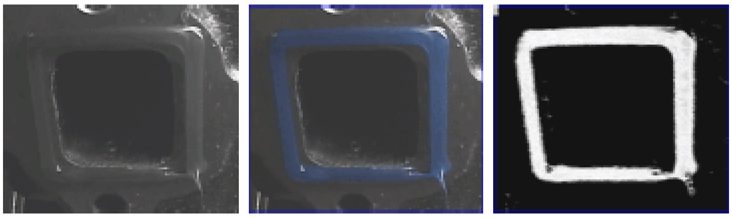


1.2.6 Color Processing / Color Recognition

Processing of color images in NeuroCheck is based on an intelligent matching process which can be customized for the specific requirements of each application. This allows color images to be analyzed with field-proven gray level algorithms and the speed and ease-of-use customary in NeuroCheck. This is especially interesting, if

- Objects have to be distinguished based on their color, e.g. electronic components or LEDs.
- Segmentation of the objects is not possible in the gray scale range because they do not exhibit a brightness contrast.

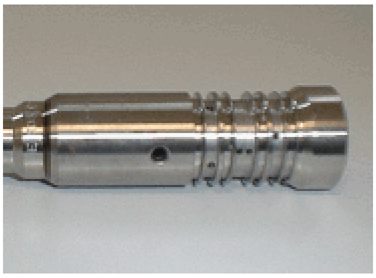
An example for the second case is shown below. The presence of sealing agent is to be tested. It hardly differs in brightness from the surface of the test piece, as shown in the left image. A color image (center, as can be seen in the online version of this manual) allows for a clear distinction which can be exploited by the NeuroCheck conversion algorithm whose result is shown on the right.

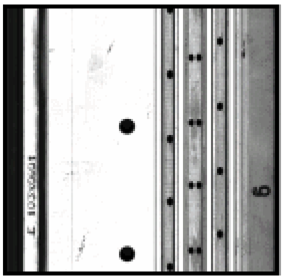


1.2.7 Surface Inspection

Surface inspection applications frequently use line-scan cameras to scan the surface of large or cylindrical parts in order to check for scratches, holes, rough areas etc.

Using linescan cameras in NeuroCheck is just as simple as using area-scan cameras. The same image processing functionality is available, shown here for gauging and defect detection on the surface of a cylindrical part.

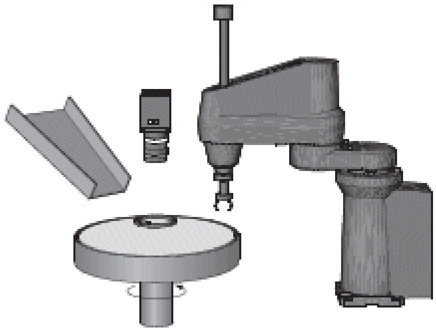




1.2.8 Position verification and robot guidance

This topic summarizes all applications in which an image processing system is used to determine the position and orientation of an object and outputs this position in order to enable another machine to handle the object.

A robot gripper can thus be enabled to grab work pieces from a conveyor belt. Another application is palletizing or depalletizing of crates containing assembly parts, packages or bottles.



1.2.9 Print Inspection

The most typical application of image comparison methods is print quality inspection. In contrast to optical character recognition, the objective here is checking the quality of the print by evaluating the correspondence of the print to a reference image.

1.3 System Requirements

Installing and using NeuroCheck requires a computer running Microsoft Windows. NeuroCheck's system requirements are the following:

- Operating system: Windows 7 (32-bit/64-bit), Vista (32-bit) or XP service pack 2 or higher (32-bit)
- Processor: 1.5 GHz (2.0 GHz multi-core recommended)
- System memory: 1 GB RAM (4 GB recommended)
- Hard disk space: 2 GB on system partition
- SVGA monitor and graphics adapter (preferred resolution at least 1024 × 768 pixels, true color)
- Drive: DVD or Blu-ray
- Interface: USB or Parallel Port

This configuration is sufficient for using NeuroCheck as a development system in an image processing laboratory. Applying NeuroCheck as a completely automatic inspection system requires the following additional NeuroCheck compatible components:

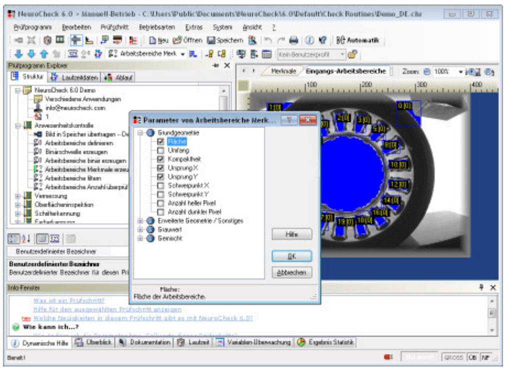
- Digital camera or frame grabber board with camera
- Digital I/O board, field bus board or serial interface for communication with PLC or master control computer

1.4 Modes of Operation

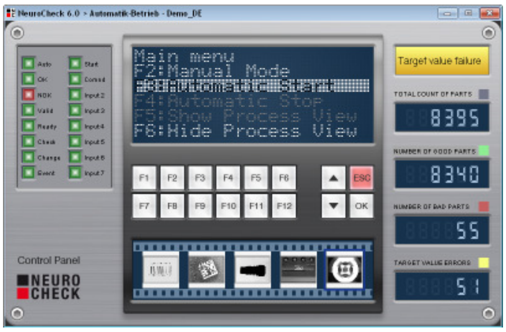
NeuroCheck integrates both the development and runtime environment. On the one hand it offers a comfortable and powerful user interface for the interactive configuration of check routines, on the other hand it contains all features necessary to use the system in an automated production environment.

That's why there are two different modes of operation in NeuroCheck, each with its own particular user interface and with its own special purpose:

Manual mode is the "development environment" and provides an user interface for check routine editing. In manual mode you construct check routines, set check function parameters and test the inspection.



Automatic mode is the "runtime environment" for executing the check routine, fully integrated into the manufacturing line. In automatic mode, NeuroCheck is typically controlled from the outside, for example by a PLC via digital I/Os or serial interface.

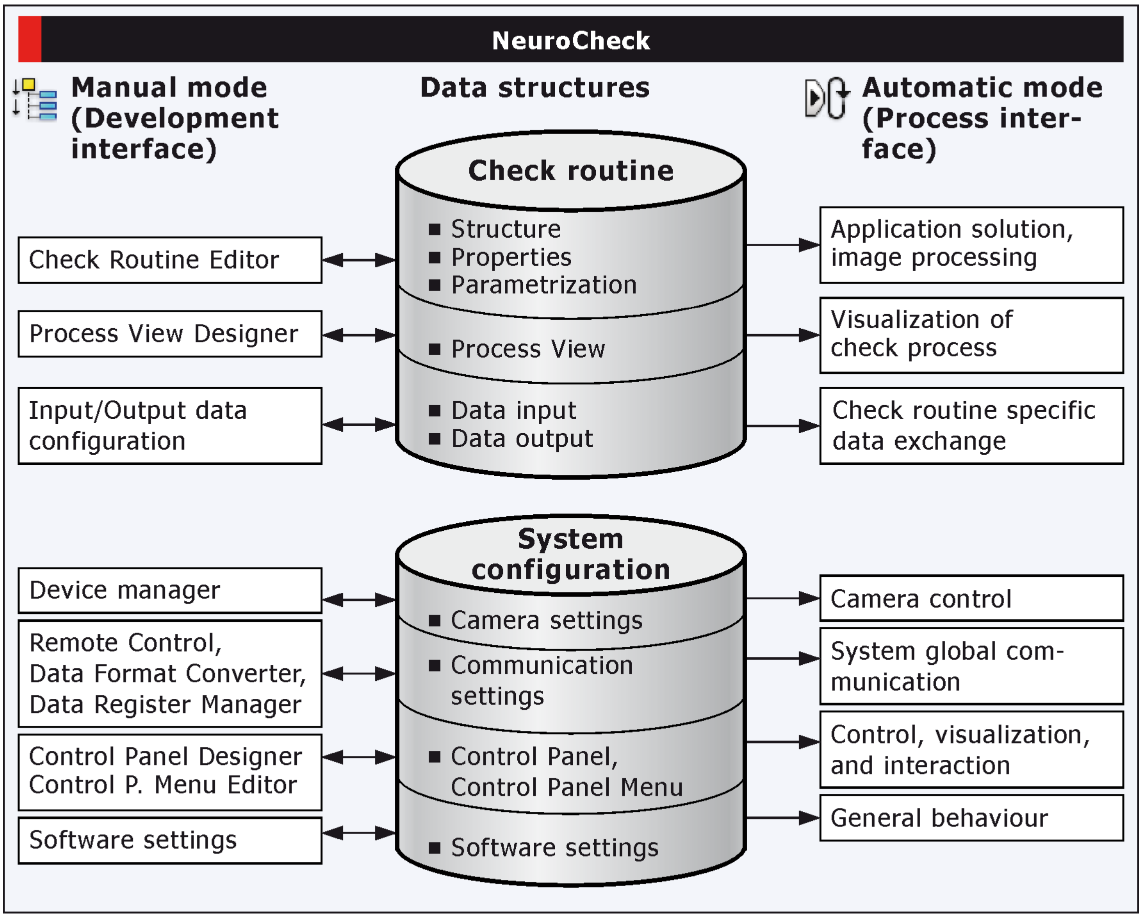


NeuroCheck has been designed to fulfill automatic visual inspection tasks integrated into modern manufacturing systems. Although it can be put to good use in a research laboratory or for interactive sample tests, where its unique user interface helps to solve varying tasks very quickly, the main focus is automatic operation on the production line. Consequently the program structure has been optimized for working in an industrial environment.

The NeuroCheck software is available in several licensing levels. These licensing levels differ in the features they provide.

1.5 Program Structure

The following figure displays the NeuroCheck software architecture from the user's perspective:



How to work with NeuroCheck

When constructing and configuring a visual inspection, you use NeuroCheck's **Manual Mode** with graphical interactive dialogs, editors and designer tools (on the left of the picture). The result of this activity are file-based data objects, especially Check Routine and Configuration files (in the center of the picture). The files determine the behaviour of NeuroCheck in **Automatic Mode** (on the right of the picture).

You can switch between these operating modes at any time. This is one of the essential strengths of NeuroCheck, especially when deploying a visual inspection system.

Check routine data

The Check Routine is the core element of every visual inspection application in NeuroCheck.

- Use the Check Routine Explorer to edit its structure, properties and parameters. Thus you are defining the core of the task's solution.
- Use the Process View Designer to define an individual visualization of the check routine.
- Use the Input Data and Output Data dialogs to define the check routine specific data to be exchanged.

Configuration data

The global configuration data determines the integration of NeuroCheck into the process environment and the overall system behaviour.

- Use the Device Manager to define the Hardware settings encompassing the configuration of both the cameras and the communication devices.
- Use the Remote Control, Data Format Converter Manager and Data Register Manager dialogs to define the communication behavior settings.
- Use the Control Panel Designer and Control Panel Menu Editor to determine the general visualization and interaction in automatic mode.
- The Software Settings define further software behaviour.

The check routine specific settings are built on top of the global settings. For example: the communication settings of the check routine specify *what* data will be transmitted, whereas the global communication settings specify *how* it will be transmitted, e.g. what protocol to use etc.

2. Check Routine

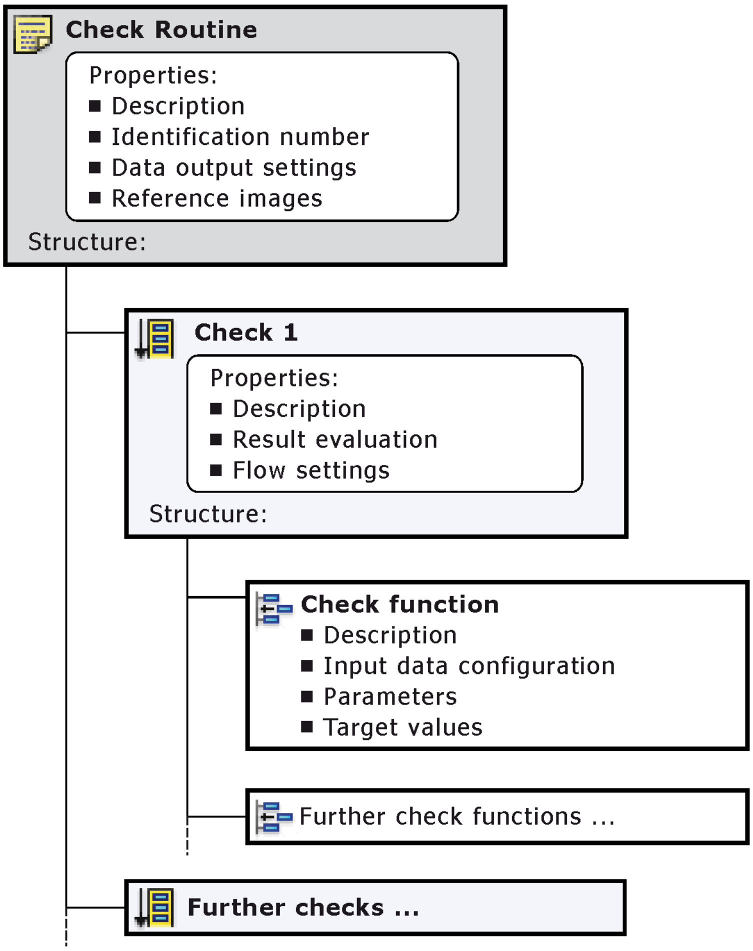
The check routine is the main element of every image processing application realized with NeuroCheck. This section describes how to create and configure check routines in manual operating mode.

2.1 The Check Routine: Introduction

The check routine is the central element of every image processing application realized using NeuroCheck. Its role can be compared to that of a text document in a word processor or a source code in a development environment.

Check routine structure

The structure of a check routine can be represented as follows:

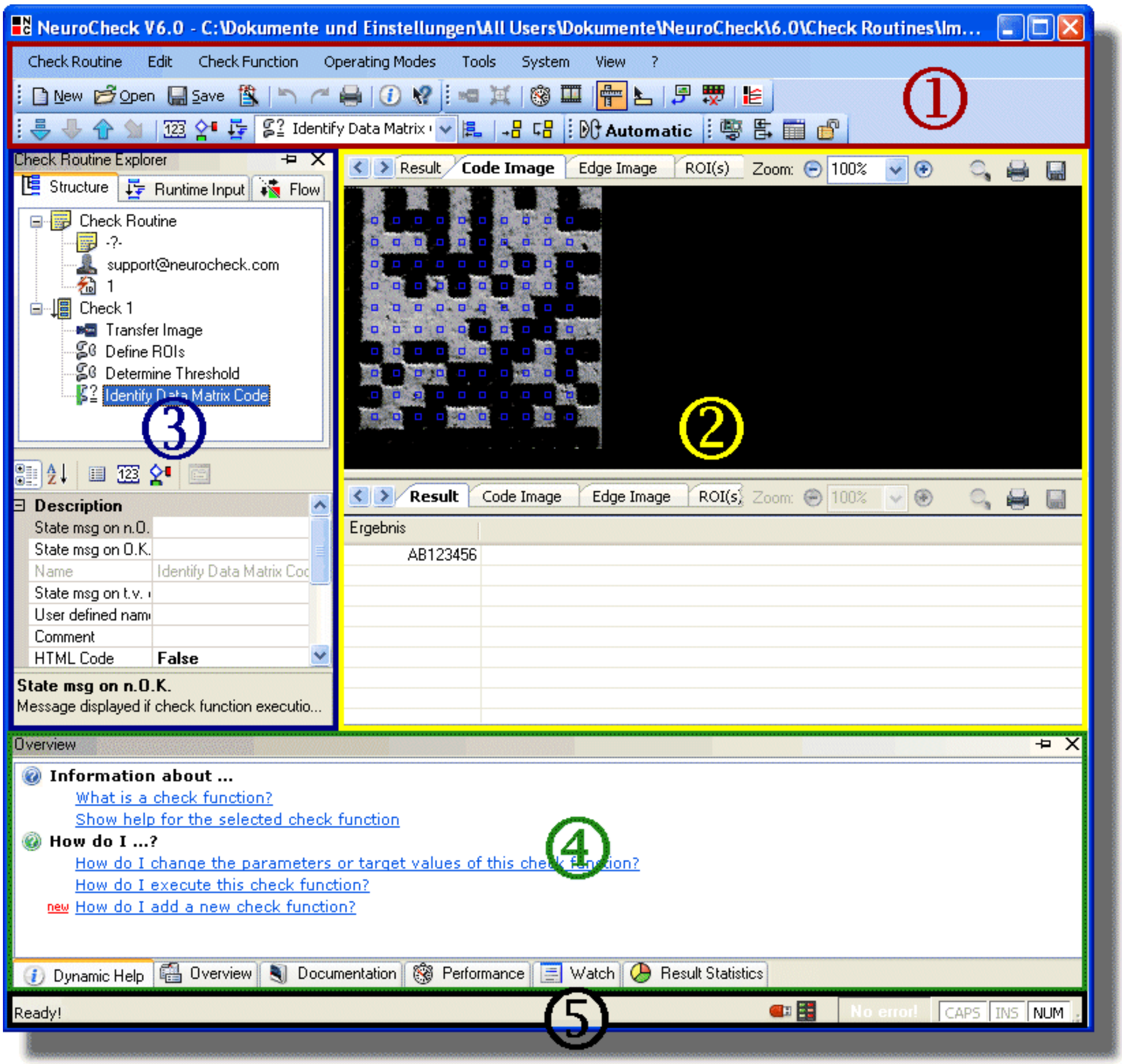


As can be seen from the figure below, a check routine is characterized on the one hand by some global properties, on the other its structure is made up of embedded structure elements. On the top level the check routine is made up of individual checks. Each check in turn has certain global properties and an internal structure made up of check functions. Each check function has its so-called input data configuration, its algorithmic parameters and its target values.

2.2 Development Interface

2.2.1 Check Routine Development Interface in Manual Mode

In operation mode **Manual** the NeuroCheck window is the development interface for your check routine. The dialog is divided into following sections:



Description:

1 Main menus and toolbars

Creating and managing check routines, configuring automatic mode, hardware configuration etc. is done using the commands from the menu or toolbars.

2 Panel for result visualizations

The **status bar** will tell you whether the hardware components necessary are available to the system or have been recognized. Status bar

3 Check Routine Explorer and Quick Edit Table

The three tabs are used for the configuration of the check routine. The **Structure** tab is the default view. The other views enable you to ensure data consistency, and to visualize or quickly access specific parameters (the latter can also be done using the Properties table of the Quick Edit Table):

- **Structure:**
  Determining each image processing step using a number of checks and check functions.
- **Runtime Input Data:**
  Editing input data configuration of check functions. This determines the data to be used by a check function.
- **Flow:**
  Editing conditional execution of checks depending on the results of other single checks.

4 Check routine Information window

This window helps you to create a check routine. Here you will get context-sensitive dynamic help, detailed information concerning input and output data, data for the documentation of your check routine etc. for your selection in the check routine explorer.

5 Status bar

The **Status bar** shows you several symbols indicating the availability and state of hardware components and other important information.

Please note that NeuroCheck usually starts up in automatic mode when it is used in visual inspection systems, so that **Manual Mode** is not available directly.

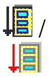
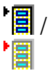
2.2.2 Check Routine Structure View

The structure view is the standard view for editing check routines. You can switch back to the structure view from any other view by clicking the tab control marked with the above icon.

The tree view

The structure tab depicts the basic structure of the check routine in the familiar tree view structure. Branches of the tree can be expanded and collapsed by double clicks or by clicking the and icons. In addition to the symbols characterizing the different NeuroCheck function groups (see section "Check function reference" for more information)...

- **… the tree view uses the following icons**

	Check routine name and description.
	Operator name.
	Check routine ID
	An individual check with its name. A red down arrow indicates that the check has been deactivated in the Check Properties dialog.
	An individual check used as a jump target (see section "Check routine control flow view" for more information on jumps). A red arrow indicates that the check has been deactivated in the Check Properties dialog.

Using the structure view

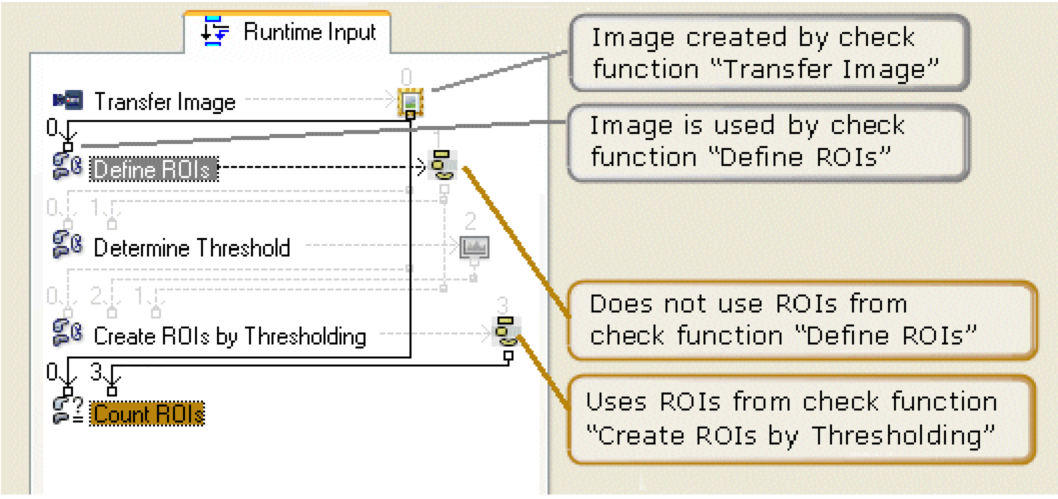
There are manifold ways of editing the check routine structure and the properties of the objects, which together form the check routine. Depending on the currently selected object different commands will be available in the editing menus (Edit and Check Function) and the Edit bar. Please refer to the respective sections for the effect of the commands. All the individual objects in the tree view as well as the view itself have context menus, containing at least a Rename or Change entry to change the designation of the object. Further items in the context menus depend on the selected object and the current state of the check routine. Please refer to the corresponding commands in section "Menus and toolbar reference" for explanations of the commands. Sometimes the Quick Edit Table may be useful.

In-place editing

The designations of all objects can be edited in place by selecting the object with a left mouse click so that it appears highlighted and clicking the object again. Pressing the ESCAPE key during editing undoes all changes, pressing the RETURN key or the left mouse button accepts the new designation.

2.2.3 Check Routine Input Data View

The input data view visualizes the data flow through the individual check which is currently selected in the Structure view. The functions forming the check are listed on the left, the objects in the data pool on the right. From each data object a line extends downwards over the whole remaining length of the check. The blue dots at the intersection points of these lines and the horizontal line above a check function indicate, which data objects are used by this check function, whereas the dotted line below a check function leads to the data object created by the function.



Types of data objects

The various types of dynamical data objects created within a check routine are depicted by the following icons:

 Image; see section "Data pool: images" for more information.

 Layer of regions of interest; see section "Data pool: regions of interest" for more information.

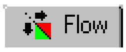
 Histogram; see section "Data pool: histogram" for more information.

 Measurement lists; see section "Data pool: measure lists" for more information.

Using the input data view

 Only a single action is possible in this view, namely to change the input data configuration of a check function, i.e. which data objects it uses. This operation is initiated by choosing **Runtime Input Data** from the Check Function menu or the context menu of the selected check function. It can also be called by clicking the corresponding icon from the Edit bar or by double-clicking the check function whose input data configuration is to be altered. All these operations open the Input Data Configuration dialog for editing the input connections of the selected check function. Section "Editing on the check function level" explains how to proceed further.

2.2.4 Check Routine Control Flow View



The control flow view visualizes the control flow between the checks in a check routine, depending on the result values of the individual checks. The control flow also affects the way the final result of the check routine is determined. For more information on this topic please refer to section "Check routine results".

Check results

A check always generates a result value of "OK" or "not OK" This result value can be treated in various ways by NeuroCheck. The result treatment can be set individually for O.K. and n.O.K. results to enable conditional branching.

Dependant on the result value the following flows are possible:

- Proceed with regular flow
- Proceed with check (GOTO)
- Jump to check (GOSUB)
- Return to calling check
- Abort check routine execution (EXIT)

Using the control flow view

The reaction of NeuroCheck to the results of an individual check is set in the Check Properties dialog for this check. The dialog is opened by selecting the check and choosing **Properties** from the Edit menu or the context menu of the check. In addition to the icons for the individual checks used in the Structure View.

2.2.5 Check Routine Result View

In manual mode, the right pane of the NeuroCheck-Window is called **Result View**. It is used to display the results of the check function currently selected in the left pane.

The Result View offers the following functionalities:

- Subdivision into two horizontal panes (by dragging the top frame downwards). Each of the panes can display a different result representation if the current check function supports several representations.
- Switching between different result representations using the register tabs in the top frame of each window pane.
- Switching between different zoom levels for each window pane using the list boxes in the top frame of each window pane.
- Saving of result images in bitmap format using the **Save** button.
- Display of the camera name, the bitmap file name or the image tray index, on the right hand side of the **Save** button.
- Copying of window content to the clipboard using one of the following methods:
    - Choose **Copy** from the context menu invoked by clicking the right mouse button inside a window pane.
    - Choose **Copy** from the **Edit** menu (copies contents of active window pane).
       Click the corresponding button on the Standard toolbar.
- Rulers can be displayed on the top and left window border for better estimation of the dimensions of image objects. Display of rulers is toggled by:
    - Choosing **Rulers** from the **View** menu.
       Clicking the corresponding button from the Extra toolbar.
- The current gray level or color value at the mouse pointer position in a displayed image can be displayed. Display of this image information is toggled by:
    - Choosing **Bitmap Info Windows** from the **View** menu.
       Clicking the corresponding button from the Extra toolbar.

2.2.6 Info Window

This window displays information about the currently selected element in the check routine tree.

You can display or hide the info window by selecting **View ▸Info window**, or you can use the pin symbol in the upper right corner to undock it and arrange it floating above the result display.

2.3 Check Routine Structure Editing

2.3.1 Check Routine Structure Editing: Introduction

The operating mode **Manual Mode** provides the development interface for creating check routines. In this mode the check routine can be edited and executed step by step manually.

Editing of the check routine structure, i.e. the sequence of checks and check functions, always takes place in the Structure view of the check routine development interface. To ensure data consistency, all other views serve only to visualize certain aspects of the check routine structure or to give more convenient access to specific parameters.

As can be seen from the check routine structure in section "Introduction to check routines" there are several structural levels in a check routine, visualized by the tree view in the Structure view.

Check routine structure editing therefore involves:

- Editing on the check routine level:
  adding and removing individual checks, start and end actions, importing checks and setting check routine properties.
- Editing on the check level:
  adding and removing check functions, adjusting the check flow and setting individual check properties.
- Editing on the check function level:
  setting check function parameters and adjusting the data flow between the check functions and .
- Editing tools:
  Several tools were designed to help you editing check routines.

Use the menu item Export as XML from the Check routine menu before and after editing check routines. The XML files created thereby make it possible to monitor the changes between different versions of check routines.

2.3.2 Editing on the Check Routine Level

2.3.2.1 Editing on the Check Routine Level

Structure editing on the check routine level involves adding and removing individual checks, importing checks and setting check routine properties. Structure editing takes place in the Structure view.

The following commands are available for structure editing on the check routine level:

- **Check Routine Wizard** from the Check Routine menu calls the Check Routine Wizard as the fastest way to generate a complete check routine structure from scratch.
- **New ▸ Check** from the Edit menu adds an empty check at the end of the check routine, immediately before the end actions if present.
- **New ▸ Start Actions** from the Edit menu adds an empty start actions check at the beginning of the check routine.
- **New ▸ End Actions** from the Edit menu adds an empty end actions check at the end of the check routine.
- **New ▸ Sub Routine** from the Edit menu creates a new sub routine.
- **Delete** from the Edit menu removes a check from the check routine, if a check is selected in the Structure View. A check cannot be removed, if it is the target of a jump (see section "Control flow view" for more information).
- **Copy Check** from the Edit menu copies the currently selected check to the end of the check routine, immediately before the end actions if present.
- **Import Check** from the Edit menu imports a check from another check routine and adds it at the end of the check routine, immediately before the end actions if present. The check to be imported is selected from the Import Check dialog box.
- **Properties** from the Edit menu opens a dialog to configure the Check routine properties.

2.3.2.2 Check Routine Wizard

2.3.2.2.1 Introduction to Check Routine Wizard

The Check Routine Wizard allows you to create Check Routines fast and easily.

Start the wizard by choosing the **Check Routine Wizard** command from the Check Routine menu or by clicking the corresponding icon on the tool bar. The pages offered by the wizard vary interactively depending on your selection.

The result of all settings made in the wizard is saved to a `*.CHR` check routine file and can be reloaded at any time.

> The wizard creates executable check routines based on bitmaps. To adjust the check routine to the current camera images of your system, conduct a manual precision adjustment (especially of check function parameters) after creating a check routine using the wizard.

2.3.2.3 Import Check

This dialog box is displayed upon choosing **Import Check** from the Edit menu and selecting a check routine in the file select dialog. It displays the individual checks of the check routine and lets you select a check to import.

The dialog box contains the following elements:

Element	Description
List view	This field displays the individual checks of the check routine together with their descriptions and commentaries. A check is selected for import by clicking the description once and then clicking **OK**.
Description	This column displays the description for the check entered in the dialog box Check Description. A mouse click on the column title sorts the checks according to the descriptions, a mouse click on one of the descriptions selects the check for import.
Commentary	This column displays the commentaries entered in the dialog box Check Description for the individual checks. A mouse click on the column title sorts the checks according to the commentaries.

2.3.3 Editing on the Check Level

2.3.3.1 Editing on the Check Level

A check (also called: individual check) is made up of check functions. In addition to setting properties of the check itself, editing on the check level therefore comprises adding and deleting check functions. Structure editing takes place primarily on the structure view.

Edit the check properties:

Properties from the Edit menu or the context menu of a check in the Structure view or the Control flow view opens the Check Properties dialog box.

To add a check function

Use the Select Check Function dialog to add a check function to the current check. The new check function can either be appended or inserted into the existing check function structure.

To remove a check function

Delete from the Edit menu removes the check function currently selected in the Structure view from the check. A check function cannot be removed, if its output data is still used by other check functions. The flow of data through the check is visualized in the Input data view. The command is also available in the context menu of the selected check.

2.3.3.2 Select Check Function

1. In the structure view of the check routine select with the left mouse button the check to which you intend to append a new check function.
2. Choose button (Add check function) from the tool bar in the upper area of the window. The dialog **Select Check Function** opens.
3. On the left side choose a check function group. (For a detailed description of all check functions corresponding to their arrangement in check function groups, please refer to "Check Function Reference").
4. Select a check function in the mid tree structure.
5. Define on the right side the position of the new check function in the check function sequence:
    - **Append Check Function**
      The selected check function will be appended to the end of this check.

   - **Insert Check Function**
     The selected check function will be inserted in this check directly before the currently selected check function.

6. Press **Close!** to exit the dialog.

Tips

- You can open the **Select Check Function** dialog by two alternative ways:
   - In the structure view, click with the **right mouse button** on the check. The context menu of the check opens in which you can choose the command **Append Check Function**.
   - Or: Choose command **New ▸Append Check Function...** from the **Edit** menu.
- If the check only contains a few or no check functions, many check functions in the **Select Check Function** dialog are marked by a red symbol because they cannot be inserted at this place.
- Please note that the input data configuration of the following check functions will not be adapted automatically to the newly inserted check function.
- In the bottom area of the dialog you find information and tips about the currently selected check function.
- The **Info** button calls the online help page for the currently selected check function.
- You may keep the dialog open and append several check functions to your check at a time.
- You can insert check functions in the check by using the mouse drag&drop operation from the **Select Check Function** dialog.

2.3.4 Editing on the Check Function Level

2.3.4.1 Editing on the Check Function Level

The check structure is affected by all parameters of its check functions. As most check function parameters represent characteristics of the individual algorithms realized by the functions, they have to be set in dialogs specific to the respective check functions. The same holds for target values. These dialogs are therefore explained in the "Check Function Reference" section, with the exception of the Input Data Configuration which is common to all check functions. This section therefore restricts itself to the common commands to use for check function editing.

With few exceptions stated below, check function editing mostly takes place in the Structure view.

There are three types of commands for check function editing:

1. Dialog commands, which actually open the dialogs used to set individual parameters.
2. Editing commands, which give you a set of useful tools when working with check function parameters.
3. Execution functions, which execute check functions so that you can see, whether they succeed in performing the appropriate operations.

More information about commands for check function editing:

- **Dialog commands**

  These commands will open dialogs for setting check function parameters, thus forming the basis for check function parameter editing. Wherever appropriate, the commands are also available in the context menu of the check function in the Structure view.

  - **Parameters** from the Check Function menu opens the parameter dialog of the check function. This dialog is specific to each function. Some check functions do not have parameter dialogs, because there are no adjustable parameters. Please refer to the section on the respective function for more information.
  - **Input Data** from the Check Function menu opens the Input Data Configuration dialog for the selected check function. Since NeuroCheck uses a unified object model for handling the data flow between check functions, this dialog is identical for all check functions.
  - **Target Values** from the Check Function menu opens the target value dialog of the check function. This dialog is specific to each function. Only so-called decision check functions do have target value dialogs, so for most check functions this command is not available. Please refer to the section on the respective function for more information.

- **Editing commands**

  The following commands are a set of useful tools for working with check function parameters.

  - **Undo** from the Edit menu reverts the most recent change made to the parameters of a check function.
  - **Redo** from the Edit menu reverts the most recent **undo** operation.
  - **Copy Parameters** from the Edit menu copies the parameter set of the check function selected in the Structure view to the internal NeuroCheck clipboard.
  - **Paste Parameters** from the Edit menu pastes the parameter set contained in the internal NeuroCheck clipboard into the check function selected in the structure view. This check function has to be of the same type as the check function, from which the parameter set had originally been copied. For example, a parameter set copied from function Create ROIs by Thresholding can only be pasted into another function Create ROIs by Thresholding (regardless of what name may currently be displayed for the function).

- **Rename** from the Edit menu lets you change the name of the check function to any string. This can also be done by editing in-place. The change can be undone by clicking button **Reset!** in the Check Function Properties dialog of the check function.
- **Display Original Check Function Name** from the Tools menu can be used to temporarily display the original names of all renamed check functions, as they appear in the Select Check Function dialog box and throughout the documentation. Executing this command a second time, the edited names will be displayed again. This command is especially useful if you have to understand the structure of a check routine written by someone else, because you can work with the familiar function names without actually having to reset them.
- **Properties** from the Check Function menu or from a check function's context menu opens the Check Function Properties dialog.
- **Help** from the Check Function menu displays the help page for the currently selected check function.

- **Execution commands**

  These commands will execute check functions so that you can see the effect of the image processing operation. All commands are also available via the context menu of the check function in the Structure view.

  - **Execute** from the Check Function executes the check up to the selected check function.
  - **Next** from the Check Function executes the function immediately following the currently selected one.
  - **Previous** from the Check Function executes the check up to the function immediately preceding the currently selected one.

> If you work with a series of bitmap files in the check function Transfer Image, there is a special additional functionality: If you hold down the Shift key while clicking on the symbol Execute, the execution will be continued with the same bitmap that is active. Otherwise, the next bitmap in the bitmap list will be taken.

2.3.4.2 Input Data Configuration

The dialog box **Input Data Configuration** is identical for all check functions because of the unified object model used by NeuroCheck to handle the data flow through a check routine. It is displayed upon choosing **Runtime Input Data** from the Check Function menu or the context menu of the check function. Alternatively you can use a double-click on the check function in the Input data view. It displays the data objects created by the check functions preceding the currently selected one within the active check. Usually the connections created automatically by NeuroCheck are correct, but special cases may require manual changes. The different data objects supported by NeuroCheck are described in section "Data Pool". The Input data view gives a comprehensive visualization of the data flow through the check using the same icons as this dialog box.

The dialog box contains the following elements:

Element	Description
Current data pool	This field displays the data objects currently available, together with name and index of the function which generated the data object. The list can be sorted by clicking the column titles.
Use	This button lets the active check function use the data object highlighted in the current data pool above as input data. The button may be unavailable if no meaningful editing operation is possible, e.g. because the selected data object is already in use.
Remove	This button removes the highlighted data object from the list of currently assigned data objects. The button may be unavailable if no meaningful editing operation is possible.
Currently assigned	This field displays the data objects currently used by the active check function, together with name and index of the function which generated the data object. The list can be sorted by clicking the column titles.
Required data	This area shows the number and types of data objects required by the active check function. You can confirm your selection with "OK" only if the number and type of objects in the "Currently assigned" field matches the objects displayed here.

2.3.5 Editing Tools

2.3.5.1 Editing Tools

There are several tools designed to help you with the editing of check routines:

- The **Stopwatch** dialog tells you how much time your check routine takes to execute. This is very important in feasibility studies, when the visual inspection has to keep to a prescribed cycle time. The dialog is opened by choosing **Stopwatch** from the **Tools** menu.
- The **Data Tray** dialog shows the current contents of the Data Tray. This dialog is opened by choosing **Data Tray** from the **Tools** menu.
- The **Image Analysis Tool** is an additional tool which is independent of any configuration in NeuroCheck. You can use it in manual mode even if there is no check routine or check function currently present. This dialog is opened by choosing **Image Analysis Tool** from the **Tools** menu.
- To avoid any problems in synchronization, you can use **Simulators** to decouple NeuroCheck from hardware components or from production line temporarily.

2.3.5.2 Stopwatch

The run time protocol allows you to determine time-intensive check functions and to optimize them (e.g. using other parameters). Open this dialog box using the **Stopwatch** command from the Tools menu or by clicking on the depicted symbol from the toolbar.

The run time protocol is updated in the background while a single check or check function is executed. For this, start the run time protocol, select a check function or a single check, and make the software execute your selection (e.g. by pressing `Ctrl+E`, clicking this symbol or **Execute** from the Check Function menu, or the context menu of the selected check function).

The display shows the run time of each executed check function and the total run time.

Please note: The total run time displayed only refers to the execution of the check function of the single checks selected in manual mode. This time is not reliable for obtaining an approximation of the execution time in automatic mode. In automatic mode, the actual time needed is slightly higher since other operations are taking place as well such as data input, data output, saving of error images, visualization, control of execution logic between single checks, time for memory allocation and release, time intervals for data exchange etc.

The stopwatch window contains the following information:

Element	Description
Name of single check	Name of single check being executed.
Sorting	Sorts the result within the check function list according to various criteria.
Check function list	If a specific check function is executed within a single check, this list contains all check functions that have been executed prior to the check function selected and that are part of the same single check. The execution time of each check function is displayed in the "Time" column.
Total time in ms	Total of run times of all check functions in list.
Advanced >> / Normal <<	This button shows or hides a graphic overview for statistic evaluation.
Sorting	Sorts the graphic overview according to various criteria.
Statistics graphs	As compared to the total execution time, this diagram shows the percentage of each check function from the list.

2.3.5.3 Data Tray Display

This dialog gives a quick overview over the contents of the global data tray.

Like the data tray itself, the display dialog for the data tray is divided into four data object types:

1. Images
2. Lists of ROIs
3. Measurement Lists
4. Histograms

The upper part of the dialog box contains an index overview for all four data types (0 - 99). Occupied spots are visually enhanced. Click an index entry to update the tab display automatically.

The middle part of the dialog box displays all currently stored data objects in the data tray in a reduced or schematic view. Click on the appropriate tab to select the desired data type. Beneath each data object, the index is displayed.

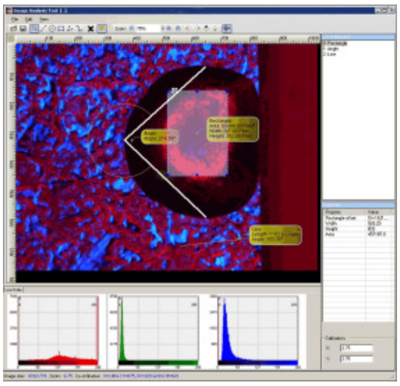
When a data object is selected, detailed information concerning this object is displayed in the lower part of the dialog box.

> The dialog is for display purposes only. Data tray access is only possible using check functions. For further information, please refer to the chapter "Data Tray: Introduction" in the function reference "Tools", sub-category "Data Tray".

2.3.5.4 Image Analysis Tool

2.3.5.4.1 Image Analysis Tool: Introduction

The image analysis tool helps you to gain information from images (e.g. for the configuration of a check routine or for project documentation purposes). The image analysis tool is an useful additional tool independent of the configuration in NeuroCheck. It can be used at any time in Manual Mode without a Check Routine having been configured.



What can you do with the Image Analysis Tool?

Gather information simply and quickly about the available images or sample parts for these purposes:

- feasibility study using sample parts or sample images
- reviewing the selected lighting arrangement
- documentation of the selected inspection strategy
- evaluation of error images

What ways of analysis are there?

- **Distance measurement** including calibration and **angle measurement**.
- For relevant sections of the image, **histograms** for the graphical display of gray level and color value distribution can be generated.
- Generate **Line Profiles** to inspect **object edges** or **brightness gradients** in the image along linear objects.
- You can display or hide **Data Tips** to display object-related measurement information.
- Inspect image areas using the **Magnifying Glass**. The magnifying glass enlarges the displayed image area around the cursor by a factor of 4.

What ways of documentation are there?

- Use individual **labels or comments,** e.g. measurements with text boxes in the image.
- **Export** images in other file formats.
- Copy images (incl. comments), histograms, line profiles or individual geometries to the **Clipboard** for use with other applications (e.g. Microsoft Office Word or project documentation).
- In JPG or TIF format, save further parameters as **EXIF information**, such as company name, author and comments in the image file's header. (EXIF = Exchangeable Image File Format).

2.3.5.5 Simulators

Editing a check routine already integrated with a production system can become a problem, especially when it uses digital I/O functions for synchronization within checks or works with many cameras, because it is not always possible to have the complete system up and running with the correct parts in front of the cameras etc. The commands **Ignore I/O** and **Simulate Image Capture** from the Tools menu temporarily decouple NeuroCheck from these hardware requirements.

The command **Use data register in Manual Mode** from the Tools menu is helpful at setting up data register connections.

Ignore I/O

Tells NeuroCheck to ignore all functions requiring signals to be received via digital I/O, field bus or other communication devices. By using this function applications remote-controlled can be edited conveniently without having to operate the whole production line with PLC etc. to generate the required signals.

These changes apply, if **Ignore I/O** is activated:

- These check functions are not executed: Read Process Input Bit, Set Process Output Bit, Read Data into Register, Send Data from Register, Control Device (for devices of category **Digital I/O**, **Field bus**, **Industrial Ethernet** or **Serial**)

Simulate Image Capture

This command temporarily deactivates all check functions that access a camera directly and switches input of all Transfer Image functions in the Check Routine to bitmap files instead of camera images. By using this function applications involving several cameras can be edited without taking care that the right piece is placed in front of each camera. Also, this mode is best suited for Error image analysis.

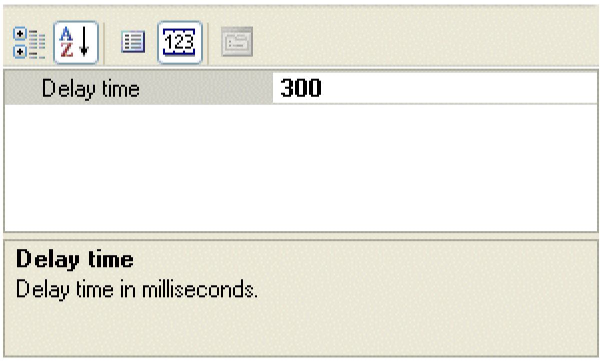
These changes apply, if **Simulate image capture** is activated:

- These check functions are not executed: **Capture Image**, **Capture Image in Parallel**, **Transfer Image to Tray**, **Control Image Acquisition**, **Control Device** (for devices of category **Camera** or **Frame grabber**).
- For check function **Transfer Image** the **image source** is switched from **camera** or **image tray** to **bitmap file**.

Use Data Register in Manual Mode

This command temporarily activates the dynamic value setting using the data registers configured in the Data Register Manager for manual mode also. In this mode the results are displayed differently to visualize the differing parameter settings used.

2.3.5.6 Quick Edit Table



The Quick Edit Table area allows for the fast editing of parameters or properties of items of the check routine tree, without the need to open the parameter or property dialogs.

You can find the Quick Edit Table in the lower area of the **Check Routine Explorer**. You can toggle the visibility of this table from the menu **View ▸ Quick Edit Table**.

2.4 Properties

This section lists the available properties of the check routine, the individual check and the check function objects.

2.4.1 Check Routine Properties

2.4.1.1 Introduction to Check Routine Properties

As a text document has such global properties like its title and author, a check routine also has properties, e.g. an identification number for automatic check routine switching. Set up these properties in the **Check Routine Properties** dialog.

The following important settings are valid for the check routine:

- **Description**:
  The check routine description consists of a name for the check routine, a comment and several other items. Some can be edited in-place in the Structure View in manual mode, others only in the corresponding property page of the Check Routine Properties dialog box.
- **Identification number**:
  To enable check routine switching by remote-control in automatic mode every check routine has an identification number. It can be edited in-place in the Structure View in manual mode or using the corresponding property page of the Check Routine Properties dialog box.
- **Write protection**:
  A check routine can be password protected to prevent changes from being made to the check routine. The write protection is activated or deactivated using the corresponding property page of the Check Routine Properties dialog box.
- **Edit Object ID**
  "Object" here simply means an element of the check routine tree, i.e. a check function, a check or the check routine itself. Each of these elements is assigned with an OID that must be unique within the current check routine. By default, the assignment of the OIDs follows a simple scheme. If necessary, the OIDs can be changed in dialog **Edit Object ID**.
- **Reference images**:
  For each camera used by a check routine, a reference image can be stored inside the check routine and used in automatic mode to verify the correct adjustment of cameras. They are explained together with the setting of camera parameters in section "Reference images".
- **Automatic mode screen configuration**:
  The display in automatic mode can be configured individually for each check routine.

2.4.1.2 General

2.4.1.2.1 Description

Here you can enter various informational items describing the check routine. Those items (displayed in the explorer of structure view) can also be edited in place directly.

The page has the following elements:

Element	Description
Check Routine Name	Here you can enter a name for the check routine to be used in the tree view and for data output, e.g. to result files.
Comment	Here you can enter an additional check routine description displayed in the tree view.
Check routine description	Here you can enter a comprehensive description of the check routine. Optionally, this description can be formatted as HTML code. An HTML description can also be written in an external HTML editor and pasted into this edit field using Ctrl+V.
Interpret as HTML code	If this option is selected, NeuroCheck will regard the operator name entered above as an email address. Provided that an email client is installed on the target machine, clicking the name in the result pane in manual mode will create an email to that address.
Operator name	Here you can enter the name of the person responsible for editing the check routine.
Interpret as e-mail address	If this option is selected, NeuroCheck will regard the operator name entered above as an email address. Provided that an email client is installed on the target machine, clicking the name in the result pane in manual mode will create an email to that address.

2.4.1.2.2 Object ID

Here you can modify the object ID of the current check routine. Each of the elements in the check routine explorer tree view is assigned an OID that must be unique within the current check routine.

The objective of this functionality is easy and unique addressing of each object by its OID. The OIDs can for instance be transmitted to XML output files. This allows the connected device to identify the source of the data within NeuroCheck.

Please don't confound the Object-ID of the check routine with the **Check Routine Identification Number** that you can edit on page **Communication** in the **Properties** dialog of the check routine.

The page contains the following elements:

Element	Description
Object ID	Here NeuroCheck displays the object identification number (OID) of the check routine. Usually you will not change this number.
Change…	Click this button to open the dialog **Edit object ID** for editing the object identification number (OID) of the selected object.

2.4.1.2.3 Test Image

Here you can enter the name of a bitmap showing, for example, a view of the test piece. This bitmap will then be displayed on a second tab page in the result pane in manual mode.

The page consists of the following elements:

Element	Description
Test piece bitmap file path	Displays path and file name of image file.
Browse…	Click this button to open a file select dialog box for selecting the bitmap to be displayed.
Reset!	This button clears the "Test piece view bitmap" name.

2.4.1.3 Automatic Mode

2.4.1.3.1 Communication

Here you can edit or enter the identification number used for automatic check routine switching.

The page contains the following element:

Element	Description
Check Routine Identification Number (CRID)	Here you can assign the identification number, by which the check routine can be called from peripherals (e.g. via Digital-I/O, Ethernet, serial interface, …). It is not recommended to use the same ID by other check routines too. Only IDs greater than 0 are allowed.

2.4.1.3.2 Result Display

Here you can configure displayed status message for the final result output of check routine.

The page contains the following elements:

Element	Description
Result O.K. / Result not O.K. Status message	Insert here the text to be displayed when the check routine yields the respective result "O.K." or "not O.K.". For instance, this text will be displayed as status message in the Result Output Window on the screen in automatic mode.
Status message for target value failure	For the case of result "not O.K." the failure can be distinguished between the cases whether the last executed check function yields a target value failure or not. Insert here the text to be displayed when the check routine was aborted due to a target value failure. This text only will be displayed if this has been configured for the respective option of the Result Output Window.

2.4.1.3.3 Process View Display

Each check routine is assigned a Process View for visualization in Automatic screen. Here you can select the layout.

The page contains the following elements:

Element	Description
Use default from global settings	With this setting, the check routine uses the Process View defined as default in Software Settings.
User defined process view	With this setting, you can select a different Process View previously created individually in Process View Designer.
Browse…	Opens a dialog to select path and file name of the Process View file linked to this check routine.

2.4.1.4 Diagnosis

2.4.1.4.1 Reference Images

Reference images offer a convenient way of checking the correct orientation of a camera in an automated visual inspection system. Maintenance work on the production line in the neighbourhood of such systems may require cameras to be temporarily removed or at least moved aside. Misalignments of cameras, defocusing or wrong aperture settings are common consequences.

NeuroCheck offers reference images to provide a way of comparing the current view of each camera with the desired view. Because the same inspection system may have to perform different checks on different work pieces, the actual image seen by the camera may differ dependent on the check routine, although the overall camera configuration stays the same. For this reason the reference images are not part of the global hardware configuration but belong to the check routine properties.

The dialog box has the following elements:

Element	Description
List of reference images	Contains a list of all currently configured reference images. They are stored as files on hard disc in the project directory.
Preview	Shows a preview of the reference image currently selected in the list.
Configure...	Opens the Reference image dialog to add and configure reference images.

Some functionality is available only if a camera is configured in the Device Manager.

2.4.2 Check Properties

2.4.2.1 Introduction to Check Properties dialog

In the **Check Properties** dialog you can configure global properties of the selected check.

The following important settings are valid for an (individual) check:

- **Description:**
  The check description consists of a name for the check, a comment and several other items. Some can be edited in-place in the Structure View in manual mode, others only in the corresponding property page of the Check Properties dialog box.
- **Type**:
  The type of the check is of prime importance for its position within the check routine. Possible types are "Standard", "Start action", "Final action" and "Sub routine"
- **Flow**:
  You can define the sequence of the check execution within the check routine using the flow settings of each individual check.

2.4.2.2 General

2.4.2.2.1 Description

Here, you can enter various informational items describing the check. Those items (displayed in the explorer of structure view) can also be edited in place directly.

The page contains the following elements:

Element	Description
Check name	This text is displayed in the Structure view behind the icon  of the check in the main window and in the dialog box Import Check.
Comment	Additional explanations about the check, e.g. its purpose and peculiarities. The commentary is also displayed in the dialog box Import Check aiding in the identification of a particular check to be imported.
Check description	This area can contain a comprehensive textual description of the individual check. Optionally, this description can be formatted as HTML code. An HTML description can also be written in an external HTML editor and pasted into this edit field using Ctrl+V.
Interpret as HTML code	If this box is checked, the description above is displayed as HTML code, else as pure text without formatting.

2.4.2.2.2 Object ID

Here you can modify the object ID of the selected individual check. Each of the elements in the check routine explorer tree view is assigned an OID that must be unique within the current check routine.

The objective of this functionality is easy and unique addressing of each object by its OID. The OIDs can for instance be transmitted to XML output files. This allows the connected device to identify the source of the data within NeuroCheck.

The page contains the following elements:

Element	Description
Object ID	Here NeuroCheck displays the object identification number (OID) of the check. Usually you will not change this number.
Change...	Click this button to open the dialog **Edit object ID** for editing the object identification number (OID) of the selected object.

2.4.2.3 Automatic Mode

2.4.2.3.1 Check type

Specify here the basic settings for the type of check and behaviour. Please note that upon changing settings in one place, settings in other places might have to be changed as well (especially Flow and Loops).

The page contains the following elements:

Element	Description
Check type	In this group you can define the check type.
Standard check	Usually single checks are of the **Default** type. In automatic mode, they are executed at their location in the check routine structure. They are usually part of the total evaluation of the check routine.
Subroutine	Subroutines are only executed if they are jumped to from a single check of the **Default** type or some other subroutine. Usually, the execution then returns to a point after the calling check. The result of a subroutine is never directly part of the total evaluation of the check routine.
Start action	The start action is a special individual check which is always executed before all the other checks. A typical use for the start action are initializations likes fast capturing of several images, which can then be stored in the image tray and processed later by individual checks.
Final action	The end action is a special individual check which is always executed after all the other checks. The end action is typically used for communication activities necessary after all image processing has been completed.
Check's result	In this group you can define how the result of the check is used in the calculation

contribution to check routine result	of the final result of the check routine.
Evaluate	The result is used in the calculation of the final result of the check routine. The final result is formed by an AND operation on all these results.
Ignore result	The result is not passed on to the check routine. If this option is selected for the "not O.K." result, the check will effectively yield "O.K." always.
Ignore result, but adopt result from subroutine	The result of this check is not transmitted to the check routine. Instead, the result of the called subroutine is subsequently inherited.
No direct contribution (subroutine)	This is the default setting for subroutines. The result of the subroutine is not transmitted to the check routine. It might have effects, nevertheless, if the selection **Adopt from subroutine** has been made in the check which jumped to this subroutine.

2.4.2.3.2 Flow

Here you can determine what check is to be executed after the current check has been executed.

A check always generates a result value of "O.K." or "not O.K." Independent from its utilization for the total result of the check routine, this result may be used to influence the flow within the check routine, for instance, to enable conditional branching. For that reason the flow settings can be set individually for "O.K." and "n.O.K." results. This is useful, if you want to apply different methods to the test piece, e.g. a fast and simple procedure and a more powerful and slower one. Another field of application are type dependent inspections.

Please note that depending on the type of the current check not all flow settings are available.

Page **Flow** can also be opened directly by using the context menu in the Control flow view.

The following settings are available for each of the possible results:

Option	Description
Proceed with regular flow (Default)	The check routine proceeds with the next check. If not explicitly configured otherwise, this means that an execution of checks from top to bottom - as the checks can be seen in the check routine structure view - takes place.
Proceed with check (GOTO)	The check routine moves to the check selected from the combo box below this option button. Afterwards, the further execution depends on the settings of the selected checks.
Jump to check (GOSUB)	The check routine jumps to the subroutine selected from the combo box below this option button. After that, the check routine flow returns to behind the jumping check.
Return to calling check	This is the default setting for subroutines. This settings enables you to jump to subroutines from various checks and ensure implicit return to the calling function.

Abort check routine execution (EXIT)	Further execution of the check routine is terminated in a controlled manner; all subsequent checks won't be executed. Exception: if there is a final action, this will always be executed.

2.4.2.3.3 Loops

Here you can configure a single check to be executed several times automatically. The loop can be terminated after a certain number of executions or after a certain result has been achieved. This can be used to create a control loop or to wait for a certain image situation to occur.

Only after all loop runs have been terminated, the execution of a single check is regarded as terminated. Then the single check result is determined. After this the execution of the single checks is continued according to the settings on the Flow page.

The page has the following control elements:

Element	Description
Loop check	If this check box is activated, the single check is executed several times in automatic mode.
Break conditions	Here you have to define one or several break conditions. After each loop, the conditions are tested for validity. If true, the loop is terminated. If not true, the loop is continued. If none of the selected break conditions occurs in automatic mode, the single check continues indefinitely. In this case NeuroCheck cannot be controlled from the outside anymore. Exception: Termination of automatic mode.
Number of loops	The loop is executed for the number of times specified, unless some other break condition occurs first.
Break on check result	The loop is terminated, if the result after loop execution is equal to the result defined here.
Result evaluation	Since the result ("O.K." or "not O.K.") after several loops can be manifold, yet a single check must report one final result, it must be defined how the final result of the single checks will be determined.
Use only last run for evaluation (default)	The final result of the single check is the result of the last loop.
"O.K." if all runs are "O.K." - otherwise "not O.K."	The final result of the single check is "O.K." if all runs are "O.K.". If at least one run is "not O.K.", the final result will be "not O.K."
"O.K." if at least one run is "O.K."	The final result of the single check is "O.K." if at least one run is "O.K.". The final result will be "not O.K." if all runs are "not O.K."

2.4.2.3.4 Deactivation

Use this function to disable a single check temporarily or for test purposes in automatic mode. This is useful, for example, if a check is used only from time to time for calibration purposes or if you want to have several versions of an inspection procedure available in a check routine.

Temporarily deactivating a check routine element harbors the risk of faulty parts going unnoticed in automatic operation. Therefore please do not use this option if you are not sure about the outcome.

The page contains the following elements:

Element	Description
Deactivate check	If this box is checked, the check will not be executed in automatic mode.
Delay execution	If the check has been deactivated, NeuroCheck will wait for the time specified in milliseconds. This option may help to solve synchronization problems occurring when checks are skipped, when the control system is expecting a specific execution time at this point.
Simulate execution	When simulation is active, the single check will act as though it has been executed. This means: • It will display a result. • This result will be used in evaluating the check routine as specified. • The result will partake in digital communication as specified. • Flow control takes place as specified. Simulation is helpful in situations where a real check is currently impossible, but the superior control system needs this check to operate.
No simulation	Default setting, no simulation takes place, i.e. the single check seemingly does not exist.
Simulate "Check O.K."	Select to simulate the result "O.K.".
Simulate "Check not O.K."	Select to simulate the result "not O.K.".

2.4.2.3.5 Result Display

Here you can configure displayed status message for the result output of check.

The page contains the following elements:

Element	Description
Result O.K. / Result not O.K. Status message	Insert here the text to be displayed when the check yields the respective result "O.K." or "not O.K.". For instance, this text will be displayed as status message in the Result Output Window on the screen in automatic mode.
Status message for target value failure	For the case of result "not O.K." the failure can be distinguished between the cases whether the last executed check function yields a target value failure or not. Insert here the text to be displayed when the check was aborted due to a target value failure. This text only will be displayed if this has been configured for the respective option of the Result Output Window.

2.4.3 Check Function Properties

2.4.3.1 Introduction to Check Function Properties Dialog

In the **Check Function Properties** dialog you can configure global properties of the selected check function.

The following important settings are valid for a check function:

- **Description:**
  The check function description consists of a name for the check function, a comment and several other items. Some can be edited in-place in the Structure View in manual mode, others only in the corresponding property page of the Check Function Properties dialog box.

2.4.3.2 General

2.4.3.2.1 Description

Here, you can enter various informational items describing the check function. Those items (displayed in the explorer of structure view) can also be edited in place directly.

The dialog box contains the following elements:

Element	Description
Check function name	Standard name of check function.

User defined name	Here you can enter an user defined name. Then, instead of the standard name, this name will be used and displayed beside the check function icon on the various views of the development interface. How do I change the name of this check function?
Check function description	This area can contain a comprehensive textual description of the selected check function. Optionally, this description can be formatted as HTML code. An HTML description can also be written in an external HTML editor and pasted into this edit field using Ctrl+V.
Interpret as HTML code	If this box is checked, the description above is displayed as HTML code, else as pure text without formatting.

To switch between the standard check function names and the user-defined ones, use from main menu Tools the Display Original Name command.

2.4.3.2.2 Object ID

Here you can modify the object ID of the selected check function. Each of the elements in the check routine explorer tree view is assigned an OID that must be unique within the current check routine.

The objective of this functionality is easy and unique addressing of each object by its OID. The OIDs can for instance be transmitted to XML output files. This allows the connected device to identify the source of the data within NeuroCheck.

The page contains the following elements:

Element	Description
Object ID	Here NeuroCheck displays the object identification number (OID) of the check function. Usually you will not change this number.
Change...	Click this button to open the dialog **Edit object ID** for editing the object identification number (OID) of the selected object.

2.4.3.3 Automatic Mode

2.4.3.3.1 Result Display

Here you can configure displayed status message for the result output of check.

The page contains the following elements:

Element	Description
Result O.K. / Result not O.K. Status message	Insert here the text to be displayed when the check function yields the respective result "O.K." or "not O.K.". For instance, this text will be displayed as status message in the Result Output Window on the screen in automatic mode or in System log window.
Status message for target value failure	For the case of result "not O.K." the failure can be distinguished between the cases whether the last executed check function yields a target value failure or not. Insert here the text to be displayed when the check was aborted due to a target value failure. This text only will be displayed if this has been configured for the respective option of the Result Output Window.

2.5 Check Routine Results

As a system for automated visual inspections, NeuroCheck is usually supposed to answer the question, whether a particular test piece is "O.K." or "not O.K.". A check routine therefore always yields one of these two results at the end of its execution.

Final check routine result

As explained in section "Introduction to check routines" a check routine is always made up of individual checks. These checks are largely independent of each other. The final result of a check routine is therefore determined from the results of each individual check – provided that this check has neither been deactivated nor been declared insignificant for the final result. There are several circumstances in which it can be useful to declare a check insignificant, e.g. if it is used solely to decide, what type of inspection has to be performed, or if in case of failure a different check has to be carried out, which then actually determines the result.

The final result of a check routine is equal to the conjunction of the results of all individual checks, which are actually carried out and whose results are not ignored.

Individual check results

A check always generates a result value of "OK" or "not OK". This result value can either be used or ignored by NeuroCheck. The treatment of the result of a check can be set using the Check Properties dialog box on page **Automatic Mode ▸Check Type**.

For the result value of an individual check the following reactions are possible:

- **Normal evaluation**:
  The result is used in the calculation of the final result of the check routine. The final result is formed by an AND operation on all these results.
- **Ignore**:
  The result of this individual check is not passed on to the check routine. If this option is selected, a possibly n.O.K. result of this individual check will not yield an n.O.K. of the check routine.
- **Ignore result, but adopt result from subroutine**:
  The result of this check is not transmitted to the check routine. Instead, the result of the called subroutine is subsequently inherited.
- **No direct contribution**:
  This is the default setting for subroutines. The result of the subroutine is not transmitted to the check routine. It might have effects, nevertheless, if the selection **Ignore result, but adopt result from subroutine** has been made in the check which jumped to this subroutine.

2.6 Data Objects

This section describes the four types of data objects that a NeuroCheck check routine uses at run-time.

2.6.1 Data Objects in Data Pool

NeuroCheck collects all data objects created during a check in the data pool. Each check function of the check can access data objects created by preceding check functions. The flow of data objects through the check functions is visualized in the **Input Data View** in manual mode. The **Input Data Configuration** dialog allows to you to select the input data for the active check function from all available data objects.

To transfer data object from one check to another, please use the Data tray.

The data objects can be of the following data types:

Images represent image information captured by a camera. Usually this will be gray level images or color images. As long as no image is available in the data pool, only check functions from the check function groups **Image acquisition**, **Tools** and **Communication** can be used in the check.

Regions of interest represent areas selected for image analysis and objects segmented in the image scene. As long as no region of interest is available in the data pool, only the check function **Define ROIs** can be used from check function group **Objects**. The region of interest is the central data type in NeuroCheck and forms the basis of NeuroCheck's flexibility.

Histograms represent information about the gray level distribution in an image. Several functions of check function group **Analysis** need this information.

Measurement lists represent geometrical information about regions of interest and their relations to each other. The check functions of check function group **Gauging** create and use lists with such measurements.

The following sections explain these data types in more detail:

- Images
- Regions of Interest
- Histograms
- Measurement Lists

2.6.2 Images

Images form the starting point of every visual inspection performed by NeuroCheck. NeuroCheck is able to process color images as well as gray value images. Images are used by most of the check functions in NeuroCheck.

An image is a numerical representation of the brightness of the captured real-world scene. A color image usually contains color information represented by the three color channels Red, Green and Blue. In a gray level image, 256 brightness levels (from 0 for Black to 255 for White) are distinguished. This is sufficient for a visually correct representation of the scene – apart from the missing color information, of course.

The functions in function group **Image preprocessing** are responsible for the processing of complete images in NeuroCheck. Most of these functions create from one (or several) images one new image of (usually) the same size, differing from the source image in some characteristics, e.g. by an enhanced contrast, reduced noise etc. Some functions do not create a new image but change the source image directly. This is stated in the documentation of the functions.

The data pool allows you to decide, which image is to be used as source image by a check function. When putting together a check from individual check functions, NeuroCheck automatically connects the check functions sequentially, i.e. a check function always uses the result image of the preceding function as its source image. The data flow is visualized in the **Input Data View** in manual mode. The **Input Data Configuration** dialog allows you to change this assignment. Thus you can filter an image in two different ways, subtract the result image from each other and go on using the result image from the subtraction – the possibilities are endless.

2.6.3 Regions of Interest

2.6.3.1 Regions of Interest: Introduction

The region of interest, short ROI, is the central data object in NeuroCheck. It describes arbitrarily shaped image regions. They are called regions of interest, because they limit the image area used for the computations performed by other functions.

The NeuroCheck region concept

NeuroCheck does not distinguish between regions manually defined by the user and regions generated automatically from image objects by segmentation or regions generated synthetically. The flexibility of NeuroCheck is based upon this essential equivalence of manual, segmented and synthetic regions. For example:

- Features, like the average gray level, and geometrical properties, like dimensions and distances, can be computed for a manually defined rectangle, e.g. for surface inspections, as well as for a segmented object.

- Segmented objects can be used as regions of interest for further segmentation stages; this means that NeuroCheck can search for image objects inside other objects as well as in rectangles or along lines.

In short: whatever can be done with one type of region can be done with the other types as well. Of course the behavior of the region types differs in some respects. It is for example perfectly possible to perform automatic position adjustment on segmented objects, but it is not meaningful, because they already are at their true positions in the image.

Region types

As said before, there are three distinct types of regions:

1. Manual regions of interest: these are defined manually by the user to tell NeuroCheck where to look for interesting things in the image (hence the name). The creation of at least one manual region of interest is necessary to use any other function from the function groups **Image analysis** and **Gauging**. This type of region is covered in detail in section "Manual regions".
2. Automatic regions of interest: these are created by check functions performing an object search within other regions. This is explained in section "Segmentation".
3. Synthetic regions of interest: these are created algorithmically based on existing regions of interest, e.g. during the computation of model geometries. This is explained in section "Synthetic regions".

Administering regions of interest

Regions of interest are organized in lists. Only a few functions in NeuroCheck actually create lists of regions of interest, other functions either alter the lists themselves (e.g. by deleting regions from the lists) or change properties of the regions in the list (e.g. by adding measurement values). Finally some functions only use the lists to compute specific values from the contents of the list (e.g. the number of regions in the list or distances between regions). In addition, the regions in each list can be organized in groups. For more information on lists and groups of regions of interest please refer to sections:

- "Lists of ROIs"
- "Groups of ROIs"

Features

Regions of Interest are often described in detail by object features. The section "Object Features" contains a detailed explanation.

2.6.3.2 Manual Regions

A manual region of interest is a prerequisite to using functions from the categories **Image Analysis** and **Gauging** as well as for some check functions from **Image Preprocessing** which are restricted to regions of interest. Manual regions are created using check function **Define ROIs**. There are these different types of manual regions: rectangle, line, polyline, circle and arc.

Rectangle

The rectangle is the most simple manual region of interest, so there is not much to say about it. Inside a rectangle NeuroCheck can search for image objects using function **Create ROIs by thresholding** or **Template Matching**. Rectangles can also be used directly by some functions like **Identify bar code** or **Unroll ROI**.

Line

A line has a starting point and an end point. Along this line the search for image objects takes place in the given direction. Lines can be filled and thus searched as an area if a surrounding area was defined before. This is configured in the check function **Define ROIs**.

For example, a line can be used for edge inspection, presence verification etc. Similar to the type **Rectangle**, the type **Line** can be used productively for almost all check functions.

Polyline

A polyline is a sequence of one or more lines optionally having a surrounding area. This area is used for clipping as explained in section "Segmentation with clipping". Polylines can be open or closed and may also be filled to be searched as an area. All these options are set using the edit field in the parameter dialog of check function **Define ROIs**.

Circle and arc

Circles and arcs can also have a surrounding area and may be filled. To generate a circular region of interest, use the circle or arc mode in the parameter dialog of check function **Define ROIs**.

(Semi-)Circular regions of interest are well suited to convert ring-shaped and curved objects into rectangular images with check function **Unroll ROI**, because compared to automatically generated objects they have a fixed length, which does not depend on illumination. This means that no interpolation is necessary to generate an unrolled image of fixed size.

An open circular region will always be unrolled from its starting point to its end point, whereas a closed circle will be unrolled in such a way that the resulting image ends at the rightmost point of the circle, i.e. at 0°.

2.6.3.3 Segmented regions

2.6.3.3.1 Segmented regions: Introduction

The term "segmented regions" refers to regions of interest created by some sort of object search within the image. An object search always starts from existing regions of interest. Therefore, a manual region has to be defined as a first step. From then on, hierarchical searches are possible with unlimited depth.

Segmentation methods

NeuroCheck supports the following segmentation methods, explained in detail in subsequent sections:

- Binary Segmentation
- Template Matching
- Edge Segmentation

2.6.3.3.2 Binary segmentation

2.6.3.3.2.1 Binary segmentation

Binary segmentation refers to the separation of objects from the image background by means of a brightness threshold. Such a threshold is computed using check function **Determine threshold**. The binary segmentation itself is done by check function **Create ROIs by thresholding**.

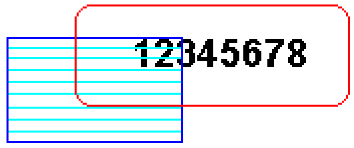
Search method

Basically, binary segmentation works as follows:

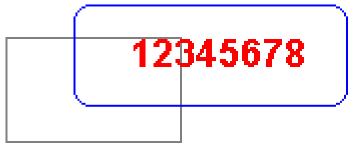
1. An existing region of interest is scanned along search rays drawn left to right across the area of the search region with an adjustable distance.
2. As soon as the gray level on the search ray changes from one side of the threshold to the other, the function knows it has just entered an object – or left, depending on the direction of the transition.
3. The function then follows the contour of the object around from the point of transition back to this point. Thus a closed object contour is created.

Examples

The following figures shows an example of a hierarchical binary search:



Search inside rectangle.



Search inside newly created region of interest.

2.6.3.3.3 Template Matching

Template matching refers to a method of finding objects based on their similarity to stored reference objects, so called templates. Such a search is conducted by check function **Template Matching**.

Search method

Basically, template matching works as follows:

1. The stored templates are moved across the search region in adjustable steps of several pixels.
2. At each step, a measure of the similarity between template and image area is computed.
3. In the neighborhood of positions with a high similarity measure, a finer search in steps of one pixel is performed.
4. The best matching positions are returned. A maximum number of positions to be returned can be set as well as a minimum similarity they have to reach.

Differences to binary segmentation

The main differences to binary segmentation are:

- Template matching does not require closed boundaries and contiguous objects. For example, it can find the letter "i" as a single object, together with its dot.
- The boundaries of regions found by template matching do not necessarily correspond to anything visible in the image. The created region of interest will always have the size and shape of the corresponding template, i.e. will always be rectangular. In fact, a region found by template matching has exactly the same properties as a manually defined rectangle (see section "Manual regions" for more information on rectangles).
- Although template matching is a two-dimensional operation by nature, function Template matching can nevertheless be applied to polylines and circles. In that case the surrounding rectangle of the polyline or circle is searched for patterns.
- The nature of template matching does not allow for clipping. If the template to be found does not fit into the search region at least once, no search is possible.

2.6.3.3.4 Edge Segmentation

Edge segmentation refers to a method of finding edge objects based on the gray level profile along a line. Such a search is conducted by check function **Create Edges**.

Search method

Basically, edge segmentation works as follows:

1. The sequence of gray levels along the search line is analyzed for transitions corresponding to adjustable slope values.
2. When such a gray level transition is found, a new region of interest is created at that location. The region will be oriented perpendicular to the search line with a fixed width.
3. The actual point of transition is computed with subpixel precision. This is very important for gauging applications. Please refer to section "Subpixel precision gauging" for more information.
4. Optionally, edges can be created in pairs, i.e. a rising edge will be considered logically connected with the next falling edge or vice versa.

Differences to binary segmentation

The main differences to binary segmentation are:

- Edge segmentation does not require fixed brightness levels of inside and outside of objects, because the slope of the gray level transition is the main criterion.
- Edge segmentation always works with subpixel precision.
- Edge segmentation is a one-dimensional operation by nature, therefore it cannot be conducted on areal regions of interest (this refers to manually defined rectangles or filled polylines as well as to regions created by template matching)

> Besides the straightforward use of edge segmentation on polylines and circles it is also possible to conduct an edge search on filled polylines and circles and on regions created by binary segmentation. In that case edges are search along the object contour.

2.6.3.4 Synthetic Regions

The term synthetic regions refers to the automatic creation of regions of interest based on existing regions by algorithmic means. This happens when group objects are created during computation of a model geometry using check function **Compute Model Geometries**. These group objects consist of the reference points of all regions in the group and lines connecting them.

2.6.3.5 Lists of ROIs

Regions of interest are created in the form of lists, possibly containing only one entry (or after certain functions, like **Screen ROIs**, not even that). Check functions can perform the following operations on lists of ROIs:

- Create lists
- Alter lists (e.g. by deleting regions from the list)
- Changing properties of regions in the list (e.g. by adding measurement values)
- Use lists (e.g. to count the regions in the list or compute distances between them)

2.6.3.6 Groups of ROIs

The Group Concept

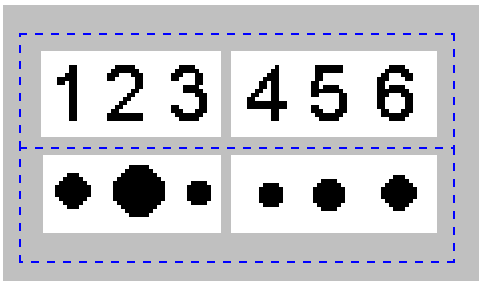
NeuroCheck is able to organize regions of interest into groups. In the default configuration all regions of interest form a single group, group 0. If, however, the check box **Group numbers** in function **Define ROIs** has been activated, the group number of each manually defined region of interest can be set individually. Group membership of every region of interest created in the course of the check depends on the group number of the manually defined region of interest forming the start of the creation chain of that particular region of interest. All regions of interest created from the same manually defined region belong to the same group in each creation step.

Manually defined regions of interest can have the same group number, although they are spatially separated. This enables identical treatment of objects found inside these regions.

Many functions from the check function groups **Image analysis** and **Gauging** allow parameter setting by groups. Groupwise parameter setting is possible only if the regions of interest in the function's input list have been created with grouping activated. In that case function parameters can be set for each group individually. In global mode all groups are treated identically.

Example

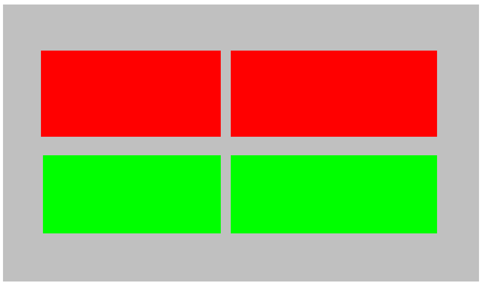
A simple example explains how to use groups in NeuroCheck. First a bitmap is loaded by check function **Transfer image**. Then two regions of interest are defined manually to make the image analysis functions available:



The check box **Group numbers** has to be activated in check function **Define ROIs**.

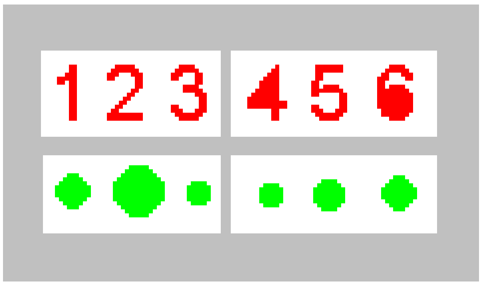
Object Search, First Step

After using check function **Determine threshold** with default parameters, a first object search can be carried out using check function **Create ROIs by thresholding**. Again the default parameters can be used, with the exception of object color, which has to be set to "Light". This function then finds the two white areas in each of the regions of interest as objects. This creates two groups of regions of interest, each containing two objects found within the same manually defined region. NeuroCheck indicates the groups by different colors:



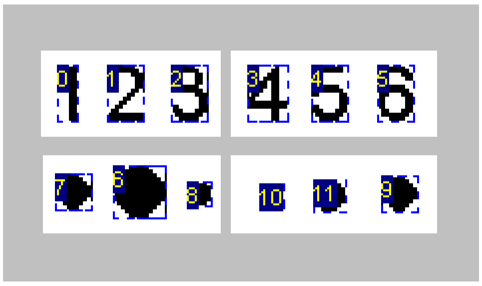
Object Search, Second Step

Now the newly created regions of interest can be searched for objects by appending the check function **Create ROIs by thresholding** again to the check, leaving all parameters at their default values. NeuroCheck then finds several objects within each of the four regions of interest. The objects in the same row belong to the same group, because their creation started within the same manually defined region of interest.



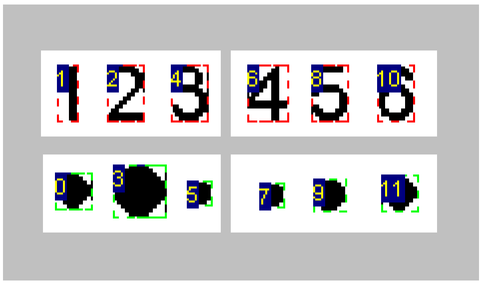
Measuring

To demonstrate the flexibility achieved by this group concept, measurements (features) are computed. In the parameter dialog of function **Compute Features**, the measurement values "Origin X" and "Area" are selected. After computing the measurements, NeuroCheck displays the regions of interest numbered in the sequence in which they had been found originally (rectangular areas are searched top to bottom and left to right).



Global Sorting

The next function to be appended to the check is **Sort ROIs**. In the parameter dialog, the sorting parameters are not set per group, but globally. The result of an ascending sort of elements 0 to 11 by "Origin X" is shown in the image below.

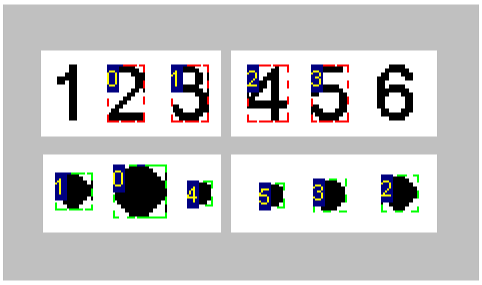


The regions of both groups form one list, which at first glance does not appear sorted at all, but indeed the sequence of the numbers agrees with the x coordinates of the enclosing rectangles of each object.

Sorting per Group

If the parameters for function **Sort ROIs** are set per group, both groups can be sorted in completely different ways.

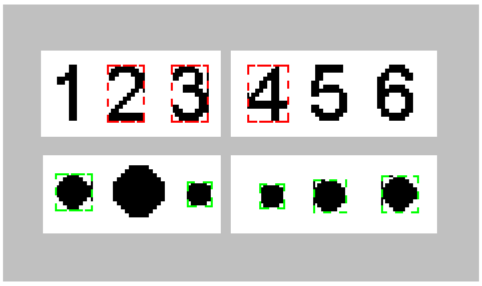
For example, it is possible to have the upper group sorted in ascending order by "Origin X", but only from element 1 to 4, the second group instead by "Area" in descending order. The result is shown in the following image.



The upper group now contains only those objects, which had the numbers 1 to 4 after the complete group had been sorted. In the lower group the largest object now has the number 0, the smallest object the number 5.

Screen Regions of Interest

Function **Screen ROIs** also allows parameters to be set per group, so that different groups can be screened according to different measurements with different minimum and maximum values. If for the upper group the permissible range of x coordinates is set from 0 to 150, only three objects of the group remain. If for the lower group an object area between 0 and 500 is allowed, the largest object will be deleted.



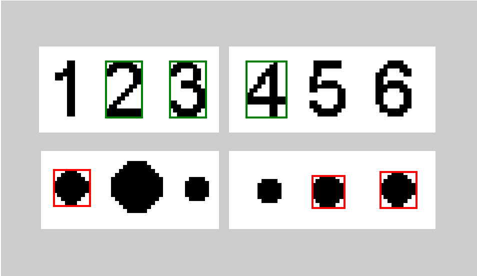
Count Regions of Interest

Function **Count ROIs** simply counts the number of all objects in the current list if the parameters are not set per group. Otherwise the number of objects in the individual groups is checked separately, yielding "OK" only, if the number is correct in every single group.

The list of regions remaining in the example after sorting and screening contains eight objects, three in the upper, five in the lower group. Assuming that a number of four or five objects is allowed in the lower group, two to three objects in the upper one, the current number could be checked by setting the global minimum of objects to six, the maximum to eight. But if the screening parameters are set as follows:

- upper group: objects with x coordinates between 0 and 150 (as before),
- lower group: objects with areas between 150 and 500,

only three objects remain in the lower group. The global counting will not detect this error, because there is still the permissible number of six objects. Setting the required number of objects individually per group detects the error. The objects in the upper group are enclosed in green frames by NeuroCheck to indicate that the group is correct, whereas the objects in the lower group are enclosed in red frames to indicate the error.



Parameter sets are assigned according to the group numbers selected in check function **Define ROIs**. Changing the group numbers there may cause inconsistencies later!

2.6.3.7 Object Features

Features are used for sorting, screening and classifying objects. Many check functions in NeuroCheck compute and use features. Those specialized to feature computation are collected in check function group **Analysis**, but there are several functions, which compute features as a byproduct of other image processing operations.

Features are stored together with regions of interest. Each region has a, possibly empty, set of features. Features can be stored in training data files and later be used by classifiers.

Features

In NeuroCheck all kinds of information describing an object, respectively a region of interest. Examples are simple scalar values computed by function **Compute features** or visual information provided by function **Resample regions of interest**. The features are stored together with the regions of interest. Subsequent check functions can use the features by accessing the corresponding list of regions of interest in the data pool.

The following check functions add features to regions of interest:

Function	Description
Compute Features	General geometrical values (like area, perimeter, width, height etc.) and gray level statistics (like average, minimum, maximum gray level and gradients).
Resample ROIs	A graphical representation of the region's enclosing rectangle with a fixed resolution in x and y direction. Most often used by check function **Classify ROIs** for character or logo recognition.

Compute Curvature	Statistical properties of the boundary of regions (like minimum, maximum, average curvature).
Template Matching	In addition to creating regions of interest based on similarity to template patterns, this function simultaneously adds several features to the detected regions: class index, class quality, X position, Y position.
Classify ROIs	Like template matching, this function adds the class index and the recognition certainty to the classified regions.
Determine Position	This function adds the computed positioning parameters to each region in its input list (even if this region is not used as a reference object for the position computation).

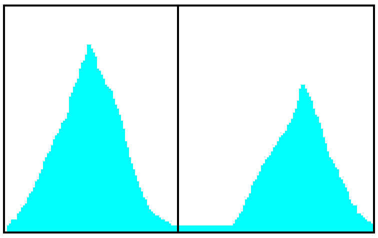
2.6.4 Histograms

A histogram is a diagram of the gray level distribution of an image region. It shows the frequency, with which each separate gray level occurs within the image region.

It can be used to compute the average brightness of an image or for the determination of thresholds. The check function **Determine threshold** analyzes the histograms of regions of interest to compute an optimal threshold for separating objects from the background. Histogram and threshold are then available to subsequent functions in the data pool, so that several functions can use the same threshold.

Basic threshold computation

The following image explains the principle of computing a threshold based on a histogram. It shows a bimodal histogram, i.e. a histogram with two distinct maxima. This property makes computing a threshold very easy. The threshold is placed at the center of gravity between the two maxima. Every image pixel, whose gray level lies above the threshold, is displayed in white, all the others black.



Check function **Determine threshold** uses only points inside of regions of interest or surrounding areas around polylines for determining the threshold. If the region consists of one line only, the number of pixels may be so small that no stable segmentation can be reached for slightly varying illumination conditions. In such a case it may be helpful to define one large region of interest for determining the threshold, then, after the threshold has been computed, several small regions for fast object search.

2.6.5 Measurement Lists

Measurement lists are usually created by check function **Gauge ROIs**. They contain geometrical measurements, which cannot be stored as properties of single image objects (i.e. regions of interest). The distance between two objects is an example for such a measurement.

Measurement lists are used by the following check functions:

Function	**creates**	**uses**	**changes**
Gauge ROIs	x		
Derive measurements	x	x	
Check allowances		x	
Combine measurement lists		x	x
Calibrate measurements		x	x

2.7 Check Function Reference

2.7.1 Check Function Groups: Introduction

The available check functions are divided into different check function groups (also called "categories"). The dialog box **Select New Check Function** contains a button for each group on the left side.

The following check function groups are available:

- Image Acquisition
- Image Preprocessing
- Objects
- Analysis
- Gauging
- Position Adjustment
- Tools
- Communication
- Plug-In

2.7.2 Image Acquisition

2.7.2.1 Introduction to Check Function Group "Image Acquisition"

Image Acquisition comprises the complete process of image capture, from the recording of a scene by the camera up to the transfer of the digitized image to computer memory for processing.

Camera

In NeuroCheck you can run digital cameras according to FireWire (IEEE 1394a/b) standard or Gigabit Ethernet cameras. In addition, you can run CameraLink cameras attached to a frame grabber board. All hardware devices for acquisition are configured in the Device Manager.

Image transmission

Usually you add check function **Capture image** to let the camera capture a new image, and then use check function **Transfer image** to make this image available in the NeuroCheck data pool for further processing.

In addition, check function group **Image Acquisition** contains further check functions for advanced tasks.

2.7.2.2 Capture Image

2.7.2.2.1 Capture Image: Introduction

Function

This function captures a new image from the selected camera. Afterwards the image has to be transferred to NeuroCheck memory for further processing using one of the functions **Transfer Image** or **Transfer Image to Tray** from the same function group.

Properties

Check function group Image Acquisition

The function has a parameter dialog.

Element	Description
Camera	Selects the camera to be used for capturing the image. The camera designations correspond to those given in the Device Manager.

Input / output data

Input	None
Output	None

How it works

Capture Image requests the digital camera or frame grabber driver to capture a new image for the selected camera. The request is handled by the driver implementation. The standard behavior of the check function is that it waits until a new image has been captured and transferred to a memory location where **Transfer Image** or **Transfer Image to Tray** will access the image data.

When NeuroCheck reaches this check function, execution is stopped and only continues

- after a new image successfully has been captured and transferred, or
- after some error occurred during digitization of the image, or
- after the time-out period specified in the Device Manager has been exceeded without successful digitization of a new image (availability dependant on driver implementation). In this case the function will return with an error, too.

> You can also apply **Capture Image** in combination with a hardware trigger for the image acquisition. From the point of view of the check function, there is no difference between a camera in free-run mode that constantly delivers new images, and a triggered camera that only delivers an image upon a hardware trigger. You just need to take care that the hardware trigger occurs while the check function is waiting for a new image, i.e. a trigger that occurred before the start or after the end of the check function will be ignored (in contrast to **Capture Image in Parallel**).
> Furthermore, a trigger also should occur within the time-out period specified in the Device Manager because the check function will abort with an error otherwise. Most driver implementations, however, allow to switch off time-out verification so that the check function will wait indefinitely long for a trigger.

2.7.2.3 Transfer Image

2.7.2.3.1 Transfer Image: Introduction

Function

This function transfers an image into the memory used by NeuroCheck. The image cannot be processed before, so this function is a prerequisite for using most of the check functions in NeuroCheck.

Properties

Check function group Image Acquisition

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

Element	Description
Save	Opens a dialog to save the current image to a file. Available file formats are `*.BMP`, `*.JPG`, `*.PNG` or `*.TIF`. Together with the option described later to use a bitmap file instead of a camera image, this allows you to interrupt your work and continue later with exactly the same image. Furthermore you can exchange images with other Windows applications.
Toolbar	Gives you the possibility to zoom and scroll the image.
Image panel	The large image field on the left displays the complete image and the section that is to be transferred to memory. Size and position of this section can be changed by mouse like a rubber band or via parameters in group box **Image section**. You can change the size of the parameter dialog by clicking and dragging the small triangle on the lower right corner of the dialog. Often it is more convenient to define the image section while having the size of the image panel enlarged that way.
Source	Select here the image source. You can select one of these options: • **Camera**: Transfer the image from camera. • **Bitmap file**: Load the image from a file (`*.BMP`, `*.JPG`, `*.PNG` or `*.TIF` format). • **Image tray**: When the image should be taken from the Data Tray. Depending on your selection, the available parameters in group box **Settings** will change.
Settings Source: Camera	**Camera:** List box from which the camera can be selected whose image is to be transferred to memory. The camera designations correspond to those in the Device Manager

dialog.

Live:
If this box is checked, the image field is continuously updated with the current camera image. This is useful for manual position adjustments, focusing etc.

New image:
Actuates updating the display with the current camera image if no live image is displayed.

Settings Source: Bitmap- File

File name:
Displays the name of the currently selected bitmap file. If more than one file has been selected in the file select box, you can switch between files in this list box.

> Later, if you are executing the sequence of check functions in manual mode, you can fix the current bitmap file of a file series by holding down the Shift key.

Browse…:
Opens a file select box for selecting one or more bitmap files to load the image from. The content of the upper list box then will be replaced by the new selection.

Settings Source: Image Tray

Image tray no.:
From this list box the number of the image to be taken from the image tray can be selected.

Show tray:
This button opens the Data Tray dialog box displaying all images available in the Data tray.

Image Section

In this section you can edit numerical values for the size and position of the selected image section. Only available, if check box **Always full image** has been deactivated.

X-Start / **Y-Start:**
Position of the upper left corner of the image section (in pixel).

Width / **Height:**
Image section size (in pixel).

Apply full image!

This button causes the complete image to be selected, without requiring manual adjustment of the section rectangle.

Always full image

If checked, each image is transferred as full image, even if the image size changes.

> This option is useful if the current image has been created by check function **Unroll ROIs**, and thus the image dimensions are variable.

Input / output data

Input	None
Output	Image with selected image section. A new data object is created in the data pool.

2.7.2.4 Capture Image in Parallel

2.7.2.4.1 Capture Image in Parallel: Introduction

Function

This function can be used to capture a new image from a camera in parallel to the running check. The digital camera or frame grabber driver through which the selected camera is accessed must be capable of parallel image acquisition upon a trigger signal and configured accordingly. Afterwards the image has to be transferred to NeuroCheck memory for further processing using one of the functions **Transfer image** or **Transfer image to tray** from the same function group. For a detailed description please refer to "How parallel image acquisition works".

Properties

Check function group Image Acquisition

The function has a parameter dialog.

Element	Description
Camera	Selects the camera to be used for capturing the image. The camera designations correspond to those given in the Device Manager.
Options	Opens the **Options for Parallel Image Acquisition** dialog (see below).

Input / output data

Input	None
Output	None

Options for Parallel Image Acquisition

This section describes the settings to be made in the options dialog of function **Capture Image in Parallel**. The dialog is opened by choosing **Options** in the parameter dialog of that function. For a more detailed description, please refer to How parallel image acquisition works.

The dialog has the following elements:

Element	Description
Stop criteria	In order to avoid an infinite wait loop when the trigger signal does not arrive, the function has several stop criteria which cause the function to stop and report an error.
Time-out	If checked, the function will wait no longer than the **Delay** time set below for the acknowledgement from the digital camera resp. frame grabber driver to arrive. After this period it will report an error. Note that the time-out from the device settings in the Device Manager will be ignored for most driver implementations.
Delay	Set here the period of time in milliseconds which the check function shall wait for the acknowledgement signal to arrive.
Behavior in case of abortion of function	This setting determines how the function behaves when one of the stop criteria will be met. There are two options: **Stop the current parallel waiting process for image trigger (Default)**: In this case, when a stop criteria is met, the function will stop the current waiting process for the image trigger. It also will stop a just started image acquisition process and discard any intermediate results. This is the default behavior of the function. **Do not stop the current parallel waiting process for image trigger, let it continue**: In this case, when a stop criteria is met, the waiting process will continue beyond the end of the check function. Thus, a just started image acquisition can continue and be finished.
Basic behavior at end of function	Here you can choose what the function will do when it ends, no matter if it ends with success or with failure (e.g. due to a stop condition). There are two options: **Automatically start a new parallel waiting process for next image trigger (Default)**: If this option is chosen,at the end of the function it will automatically start a new waiting process unless this process is already running (option **Don't stop current waiting process**, see above). This is the default behavior of the function. **Do not start a new waiting process**: If this option is chosen, the function will not start a new waiting process.

2.7.2.4.2 How parallel image acquisition works

Similar to **Capture Image**, check function **Capture Image in Parallel** requests the digital camera or frame grabber driver to capture and transfer a new image from the selected camera. The difference is, that when **Capture Image in Parallel** has detected a new image, by default it immediately starts a new image acquisition process for the digital camera or frame grabber driver before the check function returns to the NeuroCheck program flow. This acquisition process is handled by the driver implementation as an autonomous task which simply waits until the next start of an image and handles its digitization upon arrival. The main advantage is that this task is independent of the program flow in NeuroCheck so that acquisition and transfer of a new image can happen ***in parallel*** to NeuroCheck processing the previous image.

The standard application of **Capture Image in Parallel** is with an external hardware trigger signal (configured in camera or frame grabber properties in the Device Manager). This trigger signal now not only can occur while the function is waiting (like for **Capture Image**), but also while NeuroCheck is occupied with some other processing.

There are two cases for the timing:

Case 1: NeuroCheck reaches **Capture Image in Parallel** in the next inspection run before a new external trigger signal occurs.

- Detailed description

  When NeuroCheck arrives at **Capture Image in Parallel** for this case, the function cannot detect a new image and will wait for the arrival of a new image. Here, the behavior of the function is very close to that of **Capture Image**. The difference, however, is that the waiting process of **Capture Image in Parallel** can be stopped

  - after the maximum waiting time specified for the function has been exceeded,
  - upon a zero level signal of a chosen input channel,
  - by pressing the ESC key (in automatic mode only if respective feature is activated on page **Hardware** of the **Software Settings** dialog).

  If one of the above stop conditions is met, the function will end and report an error. By default, terminating the function, the parallel waiting process for a trigger signal will be stopped, which implies that it also stops the current acquisition task. Nevertheless, by default the function will immediately start a new acquisition task before returning to the NeuroCheck program flow so that the frame grabber or digital camera driver is ready for the next trigger signal. The only exceptions are when the abortion of the function is caused by ESC key (no new acquisition task started), or when special settings in the **Options** dialog are chosen, see Special Cases.

> For the very first run of **Capture Image in Parallel**, the function will by default always behave like in **Case 1** because previously no acquisition task has been started yet. You can overcome this problem by using **Control Image Acquisition** (see also section Start-up behavior).

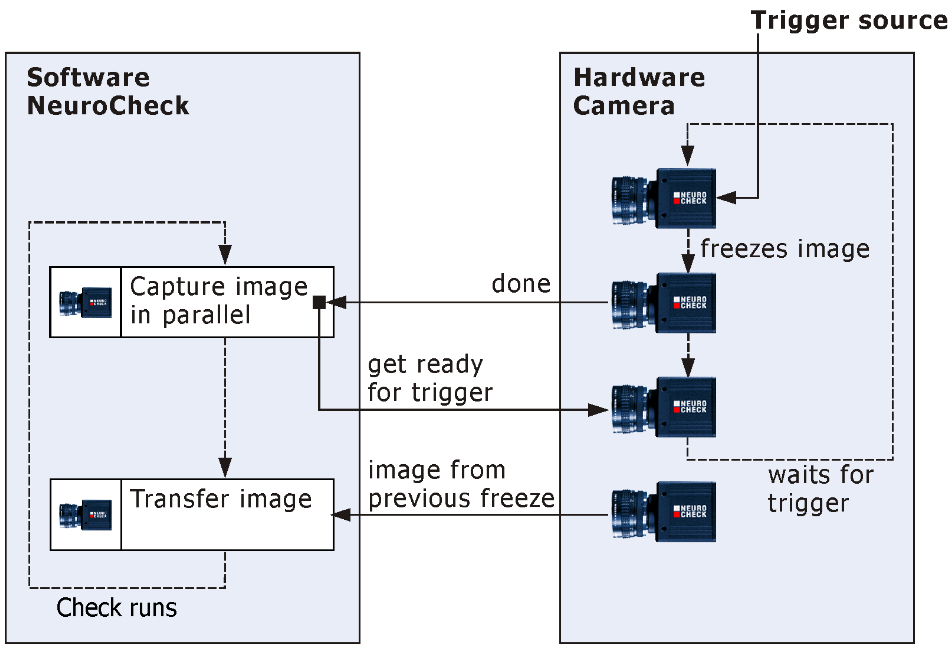
Case 2: The trigger signal occurs before NeuroCheck reaches **Capture Image in Parallel**, i.e. it occurs in parallel to the inspection in NeuroCheck.

- Detailed description

  Here you will have a real parallelism between acquisition and inspection. Thus, you can save the time necessary for an image transfer because it happens in parallel to the running check. When **Capture Image in Parallel** is reached for the next inspection run, the function will detect that a new image already is present. Then it only will start the new acquisition task and immediately return to the main program flow of NeuroCheck.

> Note that if two or more trigger signals arrive before **Capture Image in Parallel** is reached the next time, then a standard driver implementation will only use the first one and ignore all further trigger signals. As a consequence, your trigger frequency must never exceed the execution time between two image acquisitions within NeuroCheck. Otherwise, not all images will be inspected.

The following figure illustrates the operation of **Capture Image in Parallel** for this case from the second inspection run onwards. Note the parallel execution of the check itself and the image capturing.



2.7.2.4.3 Start-up behavior of Capture Image in Parallel

When **Capture Image in Parallel** is executed the first time, there is no acquisition task running yet by default. So it will start the acquisition task but also will have to wait for the new image to arrive. As a consequence it behaves differently for the first inspection run in comparison to when it is "in step" with the parallel acquisition in all further inspection runs as for Case 2 described above.

In order to overcome this problem, you can use the check function **Control Image Acquisition** with option **Initialize parallel acquisition**. This function then will start the autonomous image acquisition task for the selected camera and return immediately to the main program flow of NeuroCheck without waiting for a new image. At the same time, the function also will clear the old image for the selected camera. This would be counter-productive for a parallel operation as described in Case 2. Therefore, for the problem at hand, you should configure **Control Image Acquisition** to be executed only for the first run of the function and not for any further runs.

2.7.2.4.4 Simultaneous Acquisition from multiple Cameras

Capture Image in Parallel not only can be used to parallelize acquisition and inspection of an image in order to save execution time, but also if you wish to acquire images from several cameras at the same instant in time. In this case, you definitely have to use several instances of **Control Image Acquisition** in order to start an autonomous image acquisition process for each camera (see also section "Example 1 in Special Cases"). When all processes are running, an external trigger signal to all cameras will result in acquisition of an image for each camera at the same point in time. **Capture Image in Parallel** then still has to be used to make the different images available for subsequent functions **Transfer image** or **Transfer image to tray**.

> Please note that most frame grabber drives only support one autonomous image acquisition process per frame grabber board. So, if you wish to use **Capture Image in Parallel** with several cameras, you can only use one camera per frame grabber board even if the board type provides more than one camera input. A better alternative is to use Digital cameras because each acts as an independent device and can have its own image acquisition process.

2.7.2.4.5 Special Cases (advanced Users)

In very rare cases, it might be necessary to change the default options for starting and stopping the parallel waiting process in the Options dialog of **Capture Image in Parallel**.

Here are two examples for the advanced user:

Example 1:

- **Simultaneous Acquisition from multiple Cameras**

  Assume you have two digital cameras you wish to acquire an image from. Due to timing reasons, you want to capture both images at the same instant in time. For this, you can use an external trigger signal (e.g. from a light barrier), which will be sent to both cameras.

  If both cameras are configured for external trigger in the Device Manager, the standard setup for such an acquisition would be as follows:

  1. Control Image Acquisition for camera 1
  (initialize parallel acquisition, only first run)

  2. Control Image Acquisition for camera 2
  (initialize parallel acquisition, only first run)

  3. Capture Image in Parallel for camera 1
  (default settings)

  4. Capture Image in Parallel for camera 2
  (default settings)

  In this scenario, first cameras 1 and 2 are prepared to accept the external trigger, i.e. the parallel waiting process is started with **Control Image Acquisition** for each of them. This assures that the trigger event will be accepted by both cameras, and not only by one of them.

  Next, the first instance of **Capture Image in Parallel** will wait until the trigger event occurs and the image is completely acquired. Camera 2 will acquire its image in parallel, *at the same time* (because the same trigger event is passed on to both cameras).

  Both function **Capture Image in parallel** will start a new waiting process when they finish, so that in theory the image trigger already can occur directly after the end of **Capture Image in Parallel** for camera 2. However, this start of the parallel process also has a disadvantage:

  If there are wrong trigger events in between cycles (e.g. someone passing the light barrier while machine is stopped), then the wrong images are acquired.

  So, if you know that the image trigger cannot occur before the start of the NeuroCheck system (start signal), then you can change Control Image Acquisition for the two cameras to initialize the parallel acquisition for each inspection run. This has the advantage that the image acquisition is reset in each cycle and images acquired in between will be discarded. The disadvantage, however, is that then Control Image Acquisition has to stop the parallel waiting process for the cameras and start a new one. For some camera types this requires up to 100 ms and thus wastes valuable execution time (in most cases, the main purpose to use the simultaneous acquisition is to reduce the execution time).

So, in this case, a useful setup would be as follows

1. Control Image Acquisition for camera 1
(initialize parallel acquisition, for every run)

2. Control Image Acquisition for camera 2
(initialize parallel acquisition, for every run)

3. Capture Image in Parallel for camera 1
(Option "Do not start a new waiting process")

4. Capture Image in Parallel for camera 2
(Option "Do not start a new waiting process")

Example 2:

- **Asynchronous Lines**

Assume you have an application where you wish to inspect images from two cameras for which trigger events can occur randomly, i.e. without any synchronisation.

Given that between trigger events for each camera there is enough time, a suitable setup (simplified) could be as follows:

1. Start Actions

1. Control Image Acquisition for camera 1
(initialize parallel acquisition, only first run)

2. Control Image Acquisition for camera 2
(initialize parallel acquisition, only first run)

2. Check No. 1: "verify if image for camera 1"
(ignore "not OK", on "OK" jump to check No. 3)

1. Capture Image in Parallel for camera 1
(time-out: 1 ms, option "Don't stop current waiting process for image trigger, let it go on")

3. Check No. 2: "verify if image for camera 2"
(ignore "not OK", on "OK" jump to check No. 4)

1. Capture Image in Parallel for camera 2
(time-out: 1 ms, option "Don't stop current waiting process for image trigger, let it go on")

4. Check No. 3: "inspection of image from camera 1"

1. Transfer Image for camera 1

…

5. Check No. 4: "inspection of image from camera 2"

1. Transfer Image for camera 2

…

Starting NeuroCheck by **Timer (immediate restart)**, checks No. 1 and 2 are used to check, whether an image was captured for the respective camera or not. If a new image is ready, they jump to another check with the inspection of the captured image.

Here, for **Capture Image in Parallel**, the option "Don't stop current waiting process for image trigger, let it go on" is important. Otherwise, with such a small time-out time, it easily could happen that the check function will time out although just a trigger arrived and the image acquisition still is in progress. With the default option, the acquisition would be stopped and discarded, so that an image just would be lost!

Please note that for such an application, the result output and synchronisation with the current part is very difficult, therefore we generally would recommend to use two PC systems, one for each camera/line.

2.7.2.5 Transfer Image to Tray

2.7.2.5.1 Transfer Image to Tray: Introduction

Function

This function transfers the most recently captured image to the data tray, from where it can be transferred to memory by check function **Transfer Image** or **Copy image from tray** for further processing. This function is typically used in the Start Action of a check routine to quickly record a number of images to be processed while the test piece is already moved out of the inspection station.

Properties

Check function group Image Acquisition

The check function has parameters that can be changed dynamically.

The function has a parameter dialog.

Element	Description
Destination index	Sets the tray position on which the image will be stored. The current contents of the data tray can be viewed by choosing **Data Tray** from the **Tools** menu.
Options	Opens the **Camera Options** dialog box for selecting the camera to be used.

Input / output data	
Input	None
Output	None

2.7.2.6 Control Image Acquisition

2.7.2.6.1 Control Image Acquisition: Introduction

Function

This function controls the image acquisition from a camera by changing parameter or status settings that will be used by subsequent check functions **Capture Image** or **Capture Image in Parallel**. An instance of this check function can configured either to be executed each time it is called (like all other NeuroCheck check functions), or only for the very first call after a modus switch, i.e. after a change of check routine or operating mode. In the latter case, all subsequent calls of this instance will be ignored and the function will return immediately without any effect.

Properties

Check function group Image Acquisition

The check function has parameters that can be changed dynamically.

The function has a parameter dialog.

Element	Description
Camera	Selects the camera to be controlled. The camera designations correspond to those given in the Device Manager.
Initialize parallel acquisition	This option will initialize the parallel acquisition for the selected camera and thus prepare the usage of **Capture Image in Parallel**. This is useful if you want to perform parallel grabbing from several digital cameras or frame grabbers, or to clear images from previous runs. See the explanation of **Capture Image in Parallel** for more details.

Apply camera setup	This option can be used to apply a new camera setup to the selected camera. The settings will only be applied temporarily during run-time and will be reset for a re-start of NeuroCheck or for opening the Device Manager to the settings originally chosen there.
Browse	Opens a file select box for selecting the name of the camera setup file to be applied.
Filename	Displays the name of the currently selected camera setup file.
Execute function	Choose here if you want the function to be executed only for the first call or for each call (see above).

Input / output data

Input	None
Output	None

2.7.3 Image Preprocessing

2.7.3.1 Introduction to Check Function Group "Image Preprocessing"

Image Preprocessing usually denotes check functions, transforming an image in an image, which is basically similar to the source image, but differs in certain respects, mostly to enhance image quality. Filtering is one of the most frequently used preprocessing operations is filtering. The source image is convoluted with a filter kernel, which changes the brightness of each image pixel according to certain arithmetic rules. An example is the Mean value filter, which replaces the brightness of each pixel by the average brightness in the neighborhood of the pixel to reduce pixel noise.

Image preprocessing in NeuroCheck

Most of the preprocessing check functions in NeuroCheck affect the complete image section transferred by the **Transfer image** check function and do not need a region of interest to define their domain of influence. Some, though, are explicitly restricted to regions of interest, to increase processing speed or to perform operations specific to a region.

2.7.3.2 Filter Image

2.7.3.2.1 Filter Image: Introduction

Function

Applies a filter to the whole image. Images can be transformed in various ways by filtering. Filters are frequently applied to suppress small image degradation, to enhance edges or to emphasize particular image structures. If none of the built-in filters meets your requirements, you should select the one closest to your requirements and then manually configure it.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Input / output data

Input	Image.
Output	Filtered image. A new data object is created in the data pool.

2.7.3.2.2 Filter Image: Filter Algorithms

The filter algorithm describes the method used to derive the resulting gray level or color values from the kernel neighborhood in the source image as explained in section "Technical aspects of image filtering".

If filter mode **User defined** is selected, you can specify the filter algorithm from the list **Algorithm** in the parameter dialog of check function **Filter Image**.

The available algorithms are fundamentally divided into two groups:

1. Algorithms without weights: with these algorithms, only the presence of a point in the kernel window is important. All kernel points give the same contribution to the result value. (Shown in first table below.)
2. So-called Convolution methods give different weights to the individual kernel points. Negative values are possible, too. The gray value of each pixel is multiplied by the weight of the corresponding point in the kernel. The gray level or RGB value of the target point is determined by summing up the results of these multiplications, dividing by the sum of the kernel weights and mapping the sum to the range of gray levels respectively RGB values. The available convolution algorithms differ in the method used for this mapping. Convolution methods can be used to produce diverse results, depending on the kernel weights. It is possible to design direction sensitive gradient kernels or Gaussian low-pass filters. When a convolution method was selected, the weights can be set at will in the **Kernel shape/weights** panel. (Shown in second table below.)

The following tables describe the available algorithms:

Algorithms without kernel weights

Mean value	The mean value of all pixels in the kernel is used for the target point, smoothing the image. The mean value filter can be used for noise suppression, at the expense of blurring lines and edges.
Contrast	Computes the difference between the maximal and the minimal gray level in the kernel, to extract edges.
Rank position	The rank order of all gray levels in the kernel is computed. The gray level at the rank position entered in the Filter Options (opened by choosing the "Options" button) is used for the target point.
Median	Special case of the rank position filter, using the gray level at the medium rank position. This achieves a noise suppression with much less edge blur than the mean value filter, at the cost of higher computing time due to the costly rank ordering.
Erosion	The minimum gray level in the kernel is used for the target point, enlarging dark image structures while eliminating single bright noise pixels.
Dilation	The maximum gray level in the kernel is used for the target point, enlarging light image structures while eliminating single dark noise pixels.
Opening	Performs an erosion followed by a dilation. Thin bright lines are split open, i.e. light objects can be separated. Because two separate operations have to be carried out, the

	algorithm is comparatively slow.
Closing	Performs a dilation followed by an erosion. Thin dark lines are split open, i.e. light objects can be connected. Because two separate operations have to be carried out, the algorithm is comparatively slow.

Only the convolution methods allow for weights in the **kernel shape design panel**. Since a convolution kernel can contain negative weights, the resulting gray level for the target point can be negative, too. The available convolution algorithms differ in how they map these values to the usual gray level range of 0 to 255.

Algorithms with kernel weights (convolution)

Unsigned convolution	The absolute value of the result is used. Because in most kernels the negative weights are much weaker than the positive weights, negative results will be only slightly below zero, having a small absolute value. Therefore the result image will be rather dark. Furthermore directional information is lost, i.e. symmetrical light-dark- and dark-light-edges will yield the same result.
Signed convolution	The value of the result including its sign is mapped to a range of -128 to 127 and shifted by 128, i.e. a result of 0 will be displayed as gray level 128. The result image will be rather bright and directional information is preserved.
Positive convolution	All negative results are set to black, all positive results are used as is.
Negative convolution	The absolute values of the negative results are used as new gray levels. All positive results are set to white.

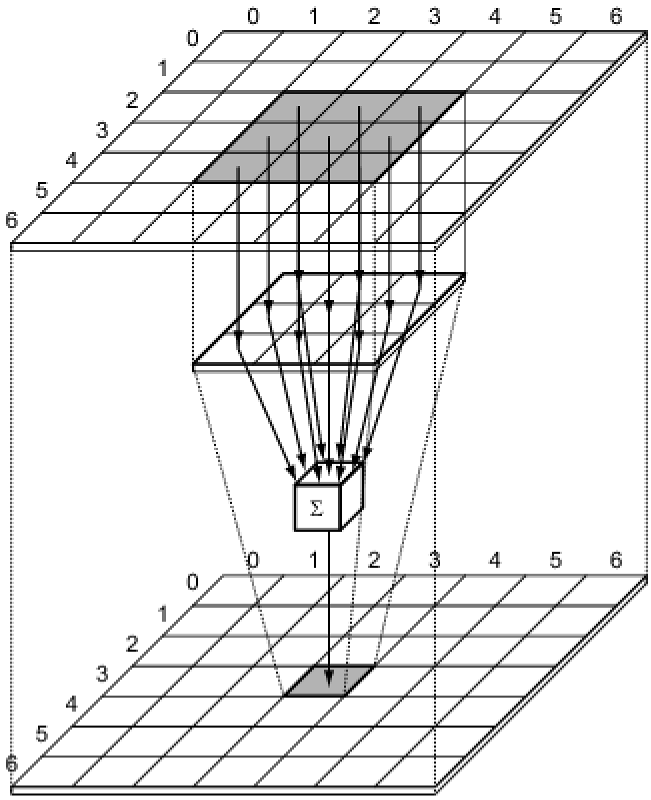
2.7.3.2.3 Technical Aspects of Image Filtering

Some theoretical background may be useful in understanding the interaction of filter kernels and algorithms. The gray level / color value of each pixel of the filtered image is computed from the gray levels / color values in a certain neighborhood of the corresponding pixel in the source image. The result of this computation, i.e. the new gray level / color value, depends on the selected neighborhood, the so-called filter kernel, and on the filter algorithm. The user can graphically design a kernel of arbitrary shape in the panel **Kernel shape/weights**. The available options for the kernel design depend on the selected algorithm.

NeuroCheck's ability to handle arbitrarily shaped filter kernels enables you to selectively enhance or suppress specified image structures. For example, it is possible by using a kernel in the shape of a diagonal line to smooth all edges running approximately along the direction of the filter kernel.

Principle of a linear filter

The following image was taken from the reference book **Industrial Image Processing** (1999, ISBN 3-540-66410-6), published by Springer Verlag (Berlin Heidelberg New York).



2.7.3.3 Copy Image

2.7.3.3.1 Copy Image: Introduction

Function

From an existing image the function creates a new image of the same size. The new image can be filled either:

1. with the contents of the source image or
2. a fixed gray level (gray level images) respectively color value (color images).

The first alternative creates an identical copy of the source image. The second alternative creates an image of uniform brightness (using gray level images) resp. of uniform color (using color images) both with the same dimensions as the source image. The function is typically used in order to preprocess an image in two different ways. The results of the different preprocessing can be combined again by using check function **Combine Images**.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Tab "Grey Level Images" (Available for gray images only)	**Copy contents**: The gray levels of the source image are copied into the new image. **Copy size and fill with gray level**: All the pixels of the new image are set to a fixed gray level. Input field **Gray level**: Specify in this box the gray level used for filling up the result image. In the field below you'll receive a preview of the selected gray level.
Tab "Color Images" (Available for color images only)	**Copy contents**: The color values of the source image are copied into the new image. **Copy size and fill with color**: All color values of the result image are set to a fix RGB value. Input field **R**, **G**, **B**: Defines the RGB color value (R = red channel, G = green channel, B = blue channel). You can define a color value either via input fields R, G and B separately or via button "..." which opens a dialog where you can select a color directly. The new selected color will be displayed in a preview field.

Input / output data

Input	Image.
Output	Copy of input image, optionally with a user-defined gray level or color value filling. A new data object is created in the data pool.

2.7.3.4 Rotate Image

2.7.3.4.1 Rotate Image: Introduction

Function

Rotates or mirrors (horizontal or vertical) the input image. Rotation is performed by multiples of 90° or by an arbitrary angle.

This check function can be used for example for the following purposes:

- Rotating the image in such a way that the inspection piece is orientated for visualization as the operator expects. This is helpful if the camera cannot be installed in the desired orientation.
- Mirroring the image so a bar code or DataMatrix code can be read correctly. This is helpful if, for example, code engraved on glass is read "from behind" and thus mirror-inverted.
- Compensation of small rotation angles to display the inspection piece in horizontal or vertical orientation. For this it is usually necessary to provide the rotation angle using a Register cell.

Properties

Check function group Preprocessing

The check function has parameters that can be changed dynamically.

The function has a parameter dialog.

Element	Description
Rotation angle	Rotates the image counterclockwise by multiples of 90°: 0°, 90°, 180° or 270°.
Free angle	Rotates the image counterclockwise by an arbitrary angle that you can edit here in the text edit box.
Options...	Opens the dialog box **Options for Rotate Image**.
Mirror horizontal	Mirrors the image along the horizontal axis (swaps top and bottom). Not available in mode **Free angle**.
Mirror vertical	Mirrors the image along the vertical axis (swaps left and right). Not available in mode **Free angle**.

Input / output data

Input	Image.
Output	Rotated or mirrored image. A new data object is created in the data pool.

2.7.3.4.2 Options for Rotate Image

This section describes the settings to be made in the options dialog of check function **Rotate image**. The dialog is opened by choosing **Options** in the parameter dialog of that check function. The dialog is only available in mode **Free angle**.

> When rotating the image around a free angle, which is not a multiple of 90°, an output image of the same size as the input image will be created. This causes two effects: Parts of the input image content is lost in the output image, and in some parts of the output image artificial image content must be created.

The dialog has the following elements:

Element	Description
Use image center	In this mode, the image will be rotated around the image center. The coordinates of the image center are displayed for your information.
Use different rotation center	In this mode, the image will be rotated using a different center of rotation whose coordinates you can enter in the boxes **X coordinate** and **Y coordinate**.
Resample method	**Nearest neighbor**: Fast execution, low quality result image. **Biquadratic**: Slow execution, high quality result image.
Gray value	Defines the gray level for the background. A preview of the selected gray level is displayed in the field below.
Red, Green, Blue	Defines the RGB-Color level for the background (R = red color channel, G = green color channel, B = blue color channel). Select this color level separately by editing the fields R, G and B or by using the button **Select color...** to open a dialog in which you can define a certain color. The defined color is displayed in a preview field.
Select color...	Opens a dialog in which you can define a certain color, alternatively to the RGB-input fields.

2.7.3.5 Combine Images

2.7.3.5.1 Combine Images: Introduction

Function

Two images of identical size are combined arithmetically. This can, for example, be used to determine the differences of two images that have been subjected to different preprocessing operations.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Addition	**Gray level images:** The gray levels of corresponding pixels were summed up. The result image will naturally be brighter than both source images, because the gray levels add up on top of each other. Gray levels are clipped to the usual range [0..255], i.e. result values exceeding 255 are set to 255 (full white). **RGB color images:** For each RGB channel, the channel color value of two input images will be added separately. The combination of the three result values builds the new RGB color value of the pixel in the result image. Because the addition of the both color values, the result image is brighter than two input images. The result value of every single channel is generally limited to range [0..255], that causes all results greater than 255 are set to the value 255.
Subtraction	The images are subtracted from each other; there are four different ways to map the possibly negative results to the gray level range resp. color range from 0 to 255: • **Absolute value**: the sign is discarded, the sequence of the images becomes irrelevant. • **Positive values only**: all negative results are set to 0 (black), the positive values are retained. • **Negative values only**: the sign of the negative results is inverted, positive results are set to 0 (black). • **Relative value**: the negative results are linearly mapped to the range 0 to 127, the positive results (including 0) to the range 128 to 255.
Mean value	The gray levels are summed up and divided by two. For color images, this is done separately for each channel; the combination of the three result values builds the new RGB color value.
Maximum	The gray level of the brighter pixel from the two images is used for the result image. For color images, this is done separately for each channel; the combination of the three result values builds the new RGB color value.

Minimum	The gray level of the darker pixel from the two images is used for the result image. For color images, this is done separately for each channel; the combination of the three result values builds the new RGB color value.

Depending on the selected operation the result image can appear quite dark because of the necessary conversion to the standard range of gray levels. This is purely a problem of display, that can be remedied by normalizing the image afterwards, but it is of no concern for subsequent image processing functions.

The check function is processing each color channel separately and subsequent to this those three values build the new result color value, so that 'Combine images' with color images can cause unexpected color changes.

Input / output data

Input	Image (source image 1). Image (source image 2).
Output	Result image. A new data object is created in the data pool.

2.7.3.6 Apply Look-up Table to Image

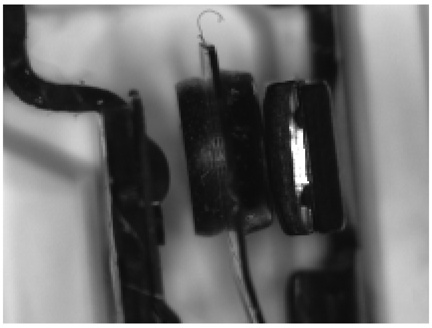
2.7.3.6.1 Apply Look-up Table to Image: Introduction

Function

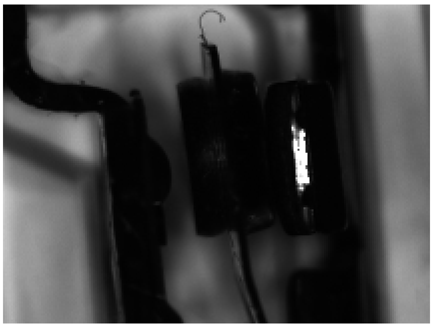
Every gray level or single color channel of the image is transformed according to the selected look-up table e.g. to emphasize certain gray level or color ranges or image structures.

Examples

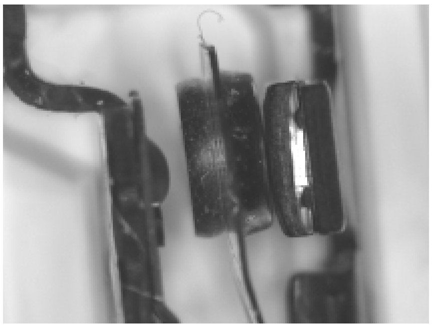
The following images are taken from the reference book **Industrial Image Processing** (1999, ISBN 3-540-66410-6), published by Springer Verlag (Berlin Heidelberg New York).



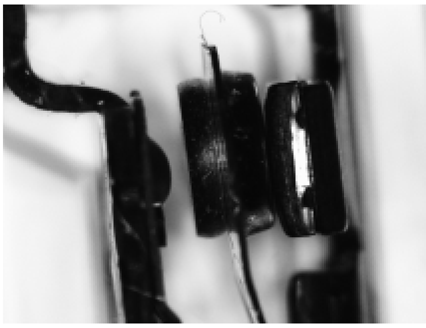
Original image



Quadratic look-up table



Square root look-up table



Sigmoid look-up table

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Gray Image	Only available for Gray level images. Here you can change the values for gray levels.
R-Channel, G-Channel, B-Channel	Only available for RGB color images. The values can be changed separately for each color channel [(R) red, (G) green and (B) blue] or for all color channels together. In the latter case, select the check box **Synchronize RGB channels** first. The statements about the computing functions of the look-up table only agree with the result image if the synchronization of the three color channels is activated. If the color channels are treated separately, it can lead to unexpected results!
Look-Up Table	In this group you can define the computing function for the look-up table.
Linear	The linear look-up table maps every gray level / color value to itself, not changing the image at all. By selecting this look-up table, the effect of the function can be temporarily neutralized without changing the structure of the check routine.
Inverse	The inverse look-up table creates a negative image.
Quadratic	The quadratic look-up table maps the gray levels / color values according to the formula ($y = x^2 / 255$). This causes a contrast enhancement: dark regions become even darker, light regions lighter.
Square root	The root look-up table uses the formula (y =sqrt (255*x)) to map the gray levels / color values. This compresses the value range, lightening dark regions and darkening light ones.
Gaussian	This s-shaped look-up table causes a sharp transition between dark and light regions not unlike a thresholding, but keeping some intermediate brightness levels.
User-defined	A user-defined look-up table is applied to the image. Using a user-defined look-up table enables you to emphasize or suppress arbitrary gray level / color value ranges. First select the function that has given the best result; then select **User-defined** to edit this function further. In user-defined mode, more settings are available:
Edit mode	The computation of the gray or color values respectively is displayed in a curve. You can edit the curve manually or as a polygon (with control points). For further information, please refer to section "User-defined look-up table". **Polygon:** The shape of the curve can be determined and changed using control points. You

	can set the number of control points in the group field **Number of control points**. **Discrete:** These values can be set separately by mouse click, e.g. to set peak points. You can also change the curve shape by dragging with the left mouse button.
Number of control points	Only available in **Polygon** mode. Here you can define whether you want the slope of the curve to have 5, 9 or 17 control points. The more control points, the more precise, for example, the computation of transitions.
Invert	Only available if **User-defined** mode was selected under **Look-up table**. Subtracts the Y-value from 255. The result is the new Y-coordinate. This means that each Y-coordinate is mirrored horizontally. Example: (0/255) results in (0/0).
Reverse	Only available if **User-defined** mode was selected under **Look-up table**. Subtracts the X-value from 255. The result is the new X-coordinate. This means that each X-coordinate is mirrored vertically. Example: (0/255) results in (255/255). It is possible to combine "Invert" and "Reverse"!

Input-/Output data

Input	Image.
Output	Gray level image with gray levels or color image with color levels mapped according to the selected look-up table. A new data object is created in the data pool.

2.7.3.6.2 User-Defined Look-Up Table

The check function enables you edit the look-up table to be applied in order to emphasize or suppress arbitrary gray level / color ranges. The look-up table is drawn with the mouse in the image panel of this dialog, keeping the left mouse button pressed. To edit the look-up table, please select the mode **User-defined** in the group box **Look-Up Table**.

Structure of a user-defined look-up table

A user-defined look-up table is represented as points in the image panel, one point for each combination of input and output gray level. The x coordinate of the point corresponds to the gray level / color value of the source image, the y coordinate gives the gray level / color value which will be created in the result image. Example: the figure may contain (amongst others) points at (10,0) and (11, 255). Every pixel with value 10 in the source image will receive value 0 in the result image, whereas every pixel with value 11 in the source image will have value 255 in the result image.

Drawing a look-up table

Some hints on drawing on "Discrete" Mode:

- Missing points are supplemented by NeuroCheck using the y coordinate of the last valid point, giving the look-up table a staircase appearance (which may be intended, though).
- Drawing a point at an x coordinate where there is a point already will delete the older point, so it is possible to refine a look-up table afterwards.
- Points can be drawn at specific position by watching the coordinate display below the diagram.
- Pressing the SHIFT key prevents the mouse from moving vertically, easing creation of the precisely horizontal lines required for stepwise look-up functions.

2.7.3.7 Enhance Image

2.7.3.7.1 Enhance Image: Introduction

Function

This check function spreads the gray levels resp. color values of the image to cover the full gray level resp. color range in order to improve image contrast or to compensate for local contrast differences due to uneven illumination. If not in "full image" mode, the image is divided into tiles (subimages) of a defined width and height. If the input image is a RGB color image, it is temporarily transformed to the HSI color space, then the intensity channel is spread, and at last the image is transformed back to RGB color space.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Size of subimages X,Y	Width (X) and height (Y) of subimages in pixels; if you enter a value larger than the respective dimension of the captured image section, the **OK** button will not be available. A mouse click on the up arrow of the size field will then set the largest accepted value.
Full image	Sets the size of the subimages to the size of the whole image; the effect is a histogram spreading for the complete image.
Background color	Please, define here the background color of the input image. • **Light**: the image has a light background (with dark objects). • **Dark**: the image has a dark background (with light objects). • **Automatic**: NeuroCheck determines the background color based on the characteristics of the image.
Gray level quantization	Adjusts the number of gray levels used in the subimages of the result image. The lowest setting creates a binary image.

Noise suppression	Influences the treatment of low contrast regions. The lowest setting enhances the contrast of these regions to the maximum value, so that image degradations become clearly visible.

Input / output data

Input	Image.
Output	Image with enhanced contrast. A new data object is created in the data pool.

2.7.3.8 Enhance Image in ROIs

2.7.3.8.1 Enhance Image in ROIs: Introduction

Function

This check function intensifies object edges within regions of interest. It converts differences in brightness in this manner to use the whole available range of brightness in the entire region of interest. Unequal contrasts or unsteady illumination can be thus corrected. If the input image is a RGB color image, it is temporarily transformed to the HSI color space, then the intensity channel is spread, and at last the image is transformed back to RGB color space.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Toolbar, Image panel, Group Parameters	Detailed Information on page "Standard parameters of check functions".
Gray level quantization	Adjusts the number of gray levels used in the subimages of the result image. The lowest setting creates a binary image.
Noise suppression	Influences the treatment of low contrast regions. The lowest setting enhances the contrast of these regions to the maximum value, so that image degradations become clearly visible.
Enhance in AOI instead of ROI	**Enabled:** The check function enhances the image in the outlining rectangle (AOI) of manually defined or automatically created regions of interest. **Disabled:**

The check function enhances the image in manually defined or automatically created regions of interest.

Input / output data

Input	Image. List of ROIs.
Output	Image with enhanced contrast. A new data object is created in the data pool.

2.7.3.9 Smooth ROIs

2.7.3.9.1 Smooth Regions of Interest: Introduction

Function

This function applies a two-stage smoothing filter to the image inside regions of interest. The main purpose of the function is to smooth ragged edges to improve edge detection.

Principle of operation

The filtering procedure is explained below using a smoothing filter in X direction with a filter length of X = 20, Y = 5, and an X-Undersampling = 10, Y-Movement = 4:

- An average (or median) row filter is moved across the image in X direction, filtering each line separately. Gray levels / color values within a 20 pixels filtering window are averaged (or their median is computed). According to the undersampling parameter for X only every 10th pixel will be used. The result is assigned to the center pixel of the filter window.
- After filtering of every image line, an average (or median) column filter is moved across the image in Y direction. The filter window contains five pixels. Because of the Movement of Y=4 , the window is shifted in steps of four pixel. According to this each result value is used four times, creating a clearly visible block structure.

If the smoothing direction is changed to Y, everything works accordingly, but the columns are filtered first in the manner described above for the lines. The essential difference is the meaning of the undersampling parameters. In smoothing direction it sets the distance of pixels within the filter window used for computing the result value, in the other direction it sets the step width for the movement of the filter window and thereby the number of times every result value is used.

Properties

Check function group Preprocessing

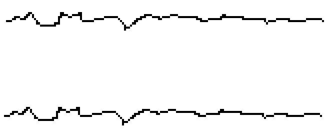
123 The function has a Parameter dialog.

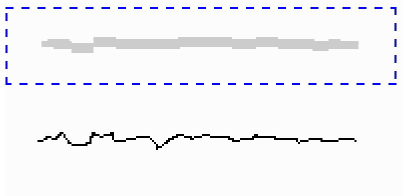
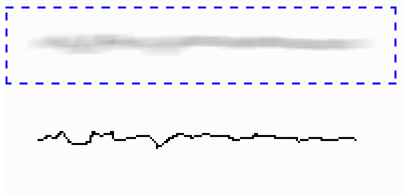
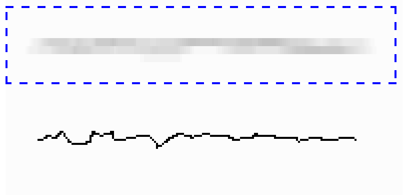
Input / output data

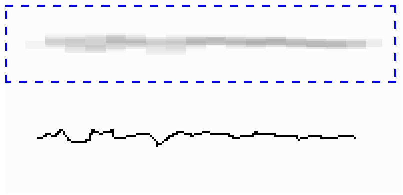
Input	Image.
Output	Smoothed Image. A new data object is created in the data pool.

2.7.3.9.2 Smooth ROIs: Examples

The examples use the following input picture:



Mode	Direction	Smoothing Length	Undersampling	Result Image
Median	X	X=20, Y=5	X-Undersampling = 10 Y-Movement = 1	
Average	X	X=20, Y=5	X-Undersampling = 10 Y-Movement = 1	
Average	X	X=20, Y=5	X-Undersampling = 10 Y-Movement = 4	

Average	Y	X=20, Y=5	X-Movement = 10 Y-Undersampling = 4	

Note the block structure created by the repetition of values through the undersampling perpendicular to the smoothing direction. Also note the angular appearance of the image filtered with the median filter.

2.7.3.10 Draw ROIs

2.7.3.10.1 Draw Regions of Interest: Introduction

Function

This check function draws regions of interest and/or the background with a RGB color value respectively gray level value. The available parameters are separated in two different tab pages, which depends on the data type of the currently used image object. The tab page "Gray level images" is not available for color images and vice versa.

The main purpose of the check function is to provide a homogeneous background or a homogeneous object interior for subsequent masking and processing steps.

Properties

Check function group Preprocessing

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

Element	Description
Tab "Gray level images"	Available for gray level images only. Contains gray level parameters for filling-up regions of interests and/or background.
Tab "Color images"	Available for color images only. Contains RGB parameters for filling-up regions of interests and/or background.

Drawing Mode	Activates the appropriate drawing mode: You can activate both checkboxes together or separately.
Fill regions	If this checkbox is activated, the check function will copy all areas outside the existing regions of interest to a new image and fill the inside of each region with the gray level, respectively color level set below.
Fill background	If this option is activated, the check function will copy the inside of all existing regions of interest to a new image and fill the background with the gray level, respectively color level set below.
Region gray level / Region color	Defines the gray level / color level which is used to paint in the image instead of the regions of interest.
Background gray level / Background color	Defines the gray level / color value which is used to paint in the image instead of the background.
Gray level	Defines the gray level. A preview of the selected gray level is displayed in the field below.
R, G, B	Defines the RGB-Color level (R = red color channel, G = green color channel, B = blue color channel). Select this color level separately by editing the fields R, G and B or by using the button '...' to open a dialog in which you can define a certain color. The new defined color displays a preview field.
Button '...'	Opens a dialog in which you can define a certain color, alternatively to the RGB-input fields.

Input / output data

Input	Image. List of regions of interest.
Output	Image with areas changed according to the parameters. A new data object is created in the data pool.

2.7.3.11 Unroll ROI

2.7.3.11.1 Unroll ROI: Introduction

Function

Transforms a stripe on both sides of the contour of a region of interest (usually a line or a circle) into a rectangular image. If the region itself is a rectangle, the function creates an image containing an exact copy of the contents of the region. The same holds for regions created by **Template Matching** as they do not have a contour corresponding to visible image features.

Where the unrolling starts depends on the type of region to be unrolled:

- For automatically generated regions (e.g. using **Create ROIs by thresholding**) unrolling starts at the topmost point of the contour.
- For manually defined **closed** regions (i.e. closed polylines and circles) unrolling **ends** at the horizontal axis, the starting point depends on the overlap (see below).
- For manually defined **open** regions (i.e. open polylines and circle sections) unrolling starts and ends at the defined end points of the polyline.

For more information on manually defined regions of interest please refer to section **Regions of interest** and check function **Define ROIs**.

Properties

Check function group Preprocessing

The check function has a parameter dialog.

- The check function has an options dialog for setting rarely needed parameters.

- The check function has an error handling dialog.

The parameter dialog has the following elements:

Element	Description
Zoom toolbar	Modifies the image display in the image panel. Detailed function description on page "Standard parameters of check functions".
Image panel	Select the region to be unrolled with the left mouse button. The surrounding area that will be unrolled is displayed here dynamically while you change the parameters to the right.
Surrounding stripe	Defines surrounding area. To display, which areas will be transferred to the result image, all ROIs of type rectangle, line and circle get a corresponding frame around their surrounding area in the image panel to the left. **Left**: Sets the width of the stripe to the left of the region of interest; for object contours

	and closed lines, left is used with respect to a clockwise movement and for open polylines with respect to the direction, in which the line has been defined. **Right**: Sets the width of the stripe to the right of the region of interest.
Overlap	This parameter is mostly used for circular regions of interest; the parameter is given in percent. For an overlap of 0%, the circle is unrolled exactly once, for an overlap of 100% exactly twice; values in between duplicate a part from the beginning of the circle to the end of the unrolled image. This can be used to avoid cutting objects in half at the point where the unrolling starts. Open polylines cannot be unrolled with overlap.
Sampling	Sets the pixel frequency for sampling the normal direction on the region of interest. Small sampling values cause the function to follow the shape of the region very closely, which may create a ragged appearance of the output image.
Smooth contour	If this check box is activated, the contour is smoothed before unrolling to reduce sampling artifacts.
Advanced...	Opens the **Options** dialog for setting parameters rarely needed.
Fixed length	If this check box is activated, the width of the result image can be set in the edit box below (this corresponds to the unrolled length). This is useful when unrolling contours of objects created by **Create ROIs by Thresholding** the length of which may vary due to illumination conditions, or polylines rotated by check function **Position ROIs**.
Fixed width	If this check box is activated, the height of the result image can be set in the edit box below. The functionality is comparable to **Fixed length**.
Error Handling	Opens a dialog to define an error handling.

Input / output data

Input	Image. List of regions of interest.
Output	Unrolled image. A new data object is created in the data pool.

2.7.3.12 Shading Correction

2.7.3.12.1 Shading Correction: Introduction

Function

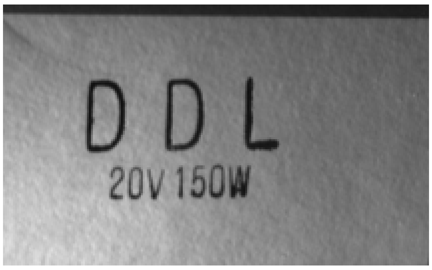
This function compensates inhomogeneous illumination conditions according to a reference image. The reference image should contain a homogeneous object, e.g. an even white surface, captured under the same illumination conditions as the current image.

The reference image has to be of the same size as the current image to provide a compensation value for each image pixel. Changing the size of the image section in check function **Transfer Image** requires a new reference image to be created.

Example:

A lamp illuminates a scene in the middle from top. Outward to the edges the image will be darker. This may cause problems on
thresholding. Using this check function, you may be able to highly attenuate such problems or to prevent them at least.

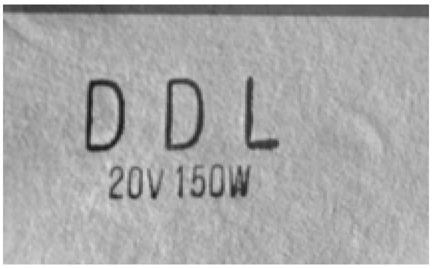
The following example was taken from the reference book **Industrial Image Processing** (1999, ISBN 3-540-66410-6), published by Springer Verlag (Berlin Heidelberg New York). It shows the effect of the shading correction on the binary thresholding.



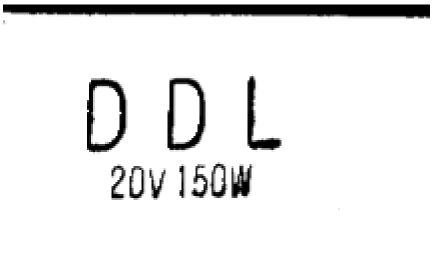
Original image

Binary image without correction

Reference image



Corrected image



Binary image after correction

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Toolbar	Detailed information on page "Standard parameters of check functions".
Current reference image	Displays the current reference image and the file name.
Teach…	Opens a second dialog to load a new reference image: **Save:** Saves the image currently displayed in a bitmap file. The contents of the displayed regions of interest will be stored as the new reference. **New image:** Causes a new image to be captured from camera, e.g. after a change of image section size. **Browse…:** Opens a file select dialog for loading an image file to be used as reference image.
Gray level offset (only on gray level images)	This value represents the actual brightness of the reference image. If an image is used as a reference image for itself, the result image will be of uniform brightness, having this gray level.
RGB color offset (only on color images)	This value represents the actual RGB color value of the reference image. If an image is used as a reference image for itself, the result image will be of uniform color, having this RGB color level.

Input / output data

Input	Gray level image or color image.
Output	Image with corrected gray levels / color values. A new data object is created in the data pool.

2.7.3.13 Filter Image on ROIs

2.7.3.13.1 Filter Image in ROIs: Introduction

Function

Applies a filter to the enclosing rectangles of manually defined or automatically created regions of interest. Filter functionality and operation are identical to function **Filter Image**.

Properties

Check function group Preprocessing

The function has a parameter dialog.

Element	Description
Image area, Zoom, Group Parameters	For information about these standard controls elements, please refer to section "Standard parameters of check functions".
Filter list	Here, you can select a filter. More information about the available filters on page "Standard filters".
Button "Modify filter parameters..."	Opens dialog **Filter Parameter**. Structure and functionality are identical to the parameter dialog of check function **Filter image**.
Filter in AOI instead of ROI	**Enabled:** The check function smoothes the image in the outlining rectangle (AOI) of manually defined or automatically created regions of interest. **Disabled:** The check function smoothes the image in manually defined or automatically created regions of interest.

Input-/output data

Input	Image.
Output	Filtered image. A new data object is created in the data pool.

2.7.3.14 Adjust Line-Scan Image

2.7.3.14.1 Adjust Line-Scan Image: Introduction

Function

Moves a configurable section from the bottom area of the image to the top. The check function is primarily used to have images obtained by rotating a work piece in front of a line-scan camera start at a defined point. See the explanation of the operation principle and the examples in the following sections.

Properties

Check function group Preprocessing

The check function has a parameter dialog.

Element	Description
360° length	Number of image rows corresponding to one full turn of the part (when scanning a rotating part using a line-scan camera). Must not exceed source image height.
Output image length	Height of the output image; must not exceed source image height. When larger than 360° length, portions of the source image will be repeated in the output image.
Manual input of start offset	If activated, the row number to be used as the first row of the output image has to be entered in the text field. The value must not exceed source image height. If not activated, a y-offset computed by check function **Determine Position** will be used as the start row.

Input / output data

Input	Image.
Output	Newly assembled image. A new data object is created in the data pool.

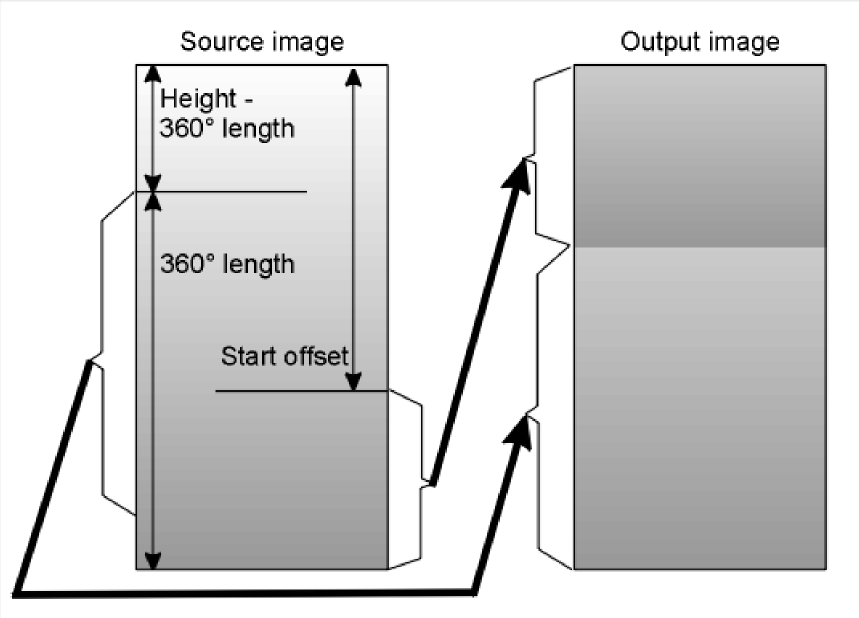
2.7.3.14.2 Adjust Line-Scan-Image: Operation

The check function **Adjust Line-Scan-Image** works as follows:

- The image area from the configured start row (or the start row derived from **Determine Position**) is to the last image row is copied from the source image to the top of the output image.
- The output image is padded up to its configured length with rows from the source image. The first row used from the source image is the row with the number "Source image height" - "360° length".

The check function is typically used for images taken by a line-scan camera from a rotating part. The output image will then contain a complete 360° unrolling beginning at a fixed angular position.

Operation principle



Explanation

The function is based on the following values:

- Source image height *H* in rows
- 360° length *L* in rows; this is the number of rows corresponding to a full rotation of the part in front of the camera. It has to be smaller or at most equal to the image height H; otherwise the function would not have sufficient rows available to represent a full rotation in the output image.
- Start position *S* in rows; this is the defined 0° position of the unrolling.

From the start position to the last image row, *(H - S)* rows remain, corresponding to *(H - S)/L * 360* degrees.

The top border of the image is H rows before that or *H/L * 360* degrees. Hence the top row corresponds to an angular position of *((H - S)/L - H/L) * 360 = -S/L * 360* degrees. Because of the 360° periodicity of the unrolling, this corresponds to *(1 - S/L) * 360 = (L - S)/L * 360* degrees in the previous rotation.

The unrolling has to be continued in the output image with the row from the previous rotation corresponding to the angular position of the bottommost row of the source image. This angular position was *(H - S)/L * 360* degree.

The angular position of a row R in the source image is *R/L * 360* degrees relative to the top image border. With the angular position of the top border of *(L - S)/L * 360* degrees this corresponds to *(R + L - S)/L * 360* degrees in the previous rotation. For a continuous unrolling, this has to be equal to *(H - S)/L * 360*. By setting these two formulas equal it follows that the rows from *R = H - L* have to be transferred from the source image to the output image.

2.7.3.15 Convert Color Image

2.7.3.15.1 Convert Color Image: Introduction

Function

This check function converts a RGB color image into a gray level image.

This check function can be used for example for the following purposes:

- Reduction of the data volume of a color image to a gray level image to accelerate subsequenced check functions that do not need color information. In that case the mode **Gray image** should be selected.
- To split off one of the three RGB channels to a new grey level image, so as to use this channel to make further analysis. In that case you can use one of the channels **Red**, **Green** or **Blue**.
- Image analysis after execution of a HSI transformation. The conversion modes **Hue**, **Saturation** and **Intensity** each generate a gray level image according to the HSI specification. The result channels of a HSI transformation provide specific analysis methods for colored regions in the image.

Properties

Check function group Preprocessing

123 The function has a parameter dialog.

Element	Description
Conversion mode	There are seven modes available to generate a result image. The modes "**Hue channel**" and "**Saturation channe**l" provide additional options.
Gray image	Converts the color image to a gray level image, factoring in all input RGB channels.
Red channel	Generates gray level image, corresponding to the content of the red channel of the input image.
Green channel	Generates gray level image, corresponding to the content of the green channel of the input image.
Blue channel	Generates gray level image, corresponding to the content of the blue channel of the input image.
Hue	Generates gray level image, corresponding to the hue channel of a HSI transformation of the color image.
Saturation	Generates gray level image, corresponding to the saturation channel of a HSI transformation of the color image.

Intensity	Generates gray level image, corresponding to the intensity channel of a HSI transformation of the color image.
Angle offset	Only available for conversion mode **Hue**. Determines the hue in the hue-color space, that is mapped to gray level 0. Used to avoid jumps in the output image between gray level 255 and 0 for the hue of the image to be analyzed later.
Low saturation for dark image areas	Only available for conversion mode **Saturation**. **Enabled:** Lower saturation will be assigned to darker image regions. **Disabled:** Dark image regions also receive the calculated value of the HSI scheme. Even minor differences in the color channels will be interpreted as major saturation.
Button "Apply!"	By choosing "Apply!" the check is executed in the background. The result is displayed in the right-hand window area.

Input-/output data

Input	Color image (RGB only).
Output	Monochrome image (1-channel-image). A new data object is created in the data pool.

2.7.3.16 Convert Image Data Type

2.7.3.16.1 Convert Image Data Type: Introduction

Function

This check function converts an image to another image with a different bit depth, for instance from 16 bit to 8 bit. This may be necessary, for example, to process an image from a 10-bit or 12-bit camera with an 8-bit algorithm.

The conversion is done for all channels independently, the check function operates independent of the color scheme of the image (gray scale image or RGB image).

Result of available conversions

- **8 bit ►16 bit**:
  The 8 most significant bits of the output image are taken from the input images, the 8 least significant bits of the output image are set to 0.
- **16 bit ►8 bit**:
  The 8 most significant bits of the input image are used as the new output image, the 8 least significant bits of the input image are deleted.
- **8 bit ►8 bit** and **16 bit ►16 bit**:
  The output image is an exact copy of the input image.

Properties

Check function group Preprocessing

The check function has a parameter dialog.

Element	Description
Current image	Here the image data type of the current input image is displayed.
Target image data type	Defines the image data type for the output image to be created.

Input / output data

Input	Image.
Output	Image with a different image data type. A new data object is created in the data pool.

2.7.3.17 Add Image Border

2.7.3.17.1 Add Image Border: Introduction

This check function adds border frames to an image.

A new output image is created, with a larger size than the input image. The enlargement width of each side can be adjusted. The contents of the input image will be copied to the output image without any change. The new color/gray values of the added border frames can be calculated using two different algorithms.

> Please note that the image content created for the border frames is artificial. This can affect the behaviour of following check functions in an unwanted way!

Properties

Check function group Image preprocessing

The check function has a parameter dialog.

Input/ output data

Input	Image.
Output	Image with added border. A new data object is created in the data pool.

2.7.3.17.2 Add Image Border: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Add Image Border**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Add Image Border** is selected.

The parameter dialog contains the following elements:

Element	Description
Border size	Defines size of each border to be added to the image. Here you can specify single border(s) or create a complete frame. **Left side:** Left border width in pixel. Range [0..999]. **Right side:** Right border width in pixel. Range [0..999]. **Upper side:** Upper border height in pixel. Range [0..999]. **Lower side:** Lower border height in pixel. Range [0..999].
Synchronize border values	**Disabled:** Separate definition of each border size. **Enabled:** Alteration of one border size will apply this value automatically to all other borders too.
Fill mode	Defines the mode how the output image data in the border area shall be generated. You can choose either a fixed filling gray / color value or the smoothing mode for the extrapolation. (View example).
Fixed value (Grey level images only!)	All new border pixels will be assigned the gray value that you can set here. **Gray value:** Select/Insert here the intensity of the gray value. Range [0..255] 0 = black, 255 = white.
Fixed value (Color images only!)	All new border pixels will be assigned the color value that you can set here. Specify the color value separately via red, green and blue value or directly via button "...". **Red:** Defines the red color channel value. Range [0..255]. **Green:** Defines the green color channel value. Range [0..255]. **Blue:** Defines the blue color channel value. Range [0..255]. **Button "...":**

	Opens a dialog in which you can define a certain color, alternative to the RGB-input fields.
Extrapolate color/gray value	The color/gray values of the new border pixels will be calculated using the outmost pixels of the input image.
Smoothing mode	The major you choose the Smoothing mode, the major will be the smoothing. **Standard:** Extrapolation of the outmost pixels only. **3x3 Smoothing:** Extrapolation of an average color /gray value, according to the border area of the input image, via quadratic form kernel size 3x3 pixels. **5x5 Smoothing:** Extrapolation of an average value, according to the border area of the input image, via quadratic form kernel size 5x5 pixels. **7x7 Smoothing:** Extrapolation of an average value, according to the border area of the input image, via quadratic form kernel size 5x5 pixels.

2.7.3.18 Resize Image

2.7.3.18.1 Resize Image: Introduction

Function

From an existing image the check function creates a new image of a different size.

This check function can be used for example in the following situations:

- You had to change the distance from the camera to the object so that the image content looks a little smaller or bigger. However, you don't want to change your check routine. Use this check function to resize your image to the original size.
- You want to classify characters, but the size of the characters in your image is too small. Use this function to make the characters a little larger.
- The pixel resolution of your camera differs between X and Y. Use this check function to resize the image in one dimension only, so that you have an approximately square pixel dimension for gauging.

> Please note that the check function is not suited for extreme changes in size. We do not recommend to use percentages smaller than 50% or greater than 200%.

Properties

Check function group Image preprocessing

The check function has a parameter dialog.

Input/ output data

Input	Image.
Output	Resized Image. A new data object is created in the data pool.

2.7.3.18.2 Resize Image: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Resize Image**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Resize Image** is selected.

The parameter dialog contains the following elements:

Element	Description
Keep width-to-height ratio	Here you can declare if you want to keep the width-to-height ratio of the input image when changing the image size.
Set default size	Resets all your settings to the size of the input image.
Absolute size in pixel	Here you can set the output image size in pixels directly.
Scaling in percent	Here you can set the output image size as percentage of the input image size. Valid range [5%..1000%].
Interpolation mode	Here you can choose the interpolation method used to create the new color/gray values. **Nearest pixel:** Fast execution, low quality result image. **Bilinear:** Slow execution, high quality result image.

2.7.4 Objects

2.7.4.1 Introduction to Check Function Group "Objects"

The check functions in group **Objects** are used to create and manipulate objects in the image. In contrast to the functions of group "Image preprocessing" they always require at least one region of interest to work with. Manual definition of a region of interest is therefore a prerequisite for using any of these check functions.

2.7.4.2 Define ROIs

2.7.4.2.1 Define Regions of Interest (ROIs): Introduction

Function

Defines Region Of Interest for subsequent check functions from check function group Analysis. Because of the importance of the ROI concept in NeuroCheck, this is one of the central functions.

Properties

Check function category Objects

The check function has parameters that can be changed dynamically.

The check functions has a Parameter dialog

- The check functions has an additional dialog to define the Error Handling.

Input/Output data

Input	Image
Output	New list of Regions Of Interest, corresponding to the defined regions. A new data object is created in the data pool.

2.7.4.3 Determine Threshold

2.7.4.3.1 Determine Threshold: Introduction

Function

This check function sets a threshold for segmenting background areas and objects. This is a prerequisite for using function **Create ROIs by thresholding**. Basic information on thresholding and the effects of several of the function parameters can be found in section "Histograms".

Properties

Check function group Objects

The function has a Parameter dialog.

The check function has parameters that can be changed dynamically.

The function can output result data. It sends resp. writes the determined threshold value.

- The function has an additional Options dialog.

Input / output data

Input	Image. List of regions of interest.
Output	Histogram with threshold (separately for each group of regions). A new data object is created in the data pool.

2.7.4.3.2 Determine Threshold: Parameter Dialog

This section describes the settings to be made in the parameter dialog of function **Determine threshold**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the function **Determine threshold** is selected.

The dialog has the following elements:

Element	Description
Automatic computation	Determines automatically an optimal threshold from the image contents within the regions of interest in its input list. For more information about threshold computation please refer to section "Introduction to histograms".
Result image	Adjusts the predominance of light or dark areas in the result image.

Defect suppression	This parameter suppresses disturbances caused by very bright or very dark image regions. If set too high, computation of a threshold may be impossible due to insufficient numbers of pixels. You will find more information about this in section "Defect suppression in histograms".
Options	Opens the **Options for Automatic Thresholding** dialog.
Manual input	If this option is selected, a global threshold can be entered manually.
Threshold	Sets the manual threshold.
Automatic readjustment	If this check box is activated, NeuroCheck automatically computes a reference gray level from the regions of interests in the current input image. For color images NeuroCheck calculates an average of all gray level values of the three color channels (red, green and blue). The manually entered threshold is then regarded as relative to this reference level, i.e. when the check routine is executed and the camera image is brighter than the current image, the threshold will be automatically increased internally and vice versa. If the check box is disabled the given threshold is treated as absolute value.
Range thresholding	If this check box is activated, NeuroCheck treats only part of the gray levels above the threshold as (logically) white. For color images the part of gray level is the average of all gray level values of the three color channels (red, green and blue). A range setting of 50% means that gray levels from the threshold up to a value halfway between the threshold and the highest possible value of 255 are regarded as white, all others as black. Please refer to section "Range thresholding" for more information.
Live preview	You can check this box to have a permanent live preview of the binary image while the dialog is open.

2.7.4.4 Create ROIs by Thresholding

2.7.4.4.1 Create ROIs by Thresholding: Introduction

Function

This function creates new regions of interest starting from existing regions of interest by a thresholding process. Therefore the check function **Determine threshold** has to be executed first. The existing regions of interest are thresholded and searched for objects using the threshold determined in advance (this can be one global threshold for all regions or an individual threshold for each group of regions). From the detected objects a new list of regions of interest is created, that can be processed further. All regions of interest created from within regions belonging to the same group will form a group with the same number in the resulting list of regions.

Properties

Check function group Objects

The function can output result data. It sends resp. writes the number of created ROIs.

The check function has a Parameter dialog.

- The check function has an additional Options dialog.

- The check function has an additional dialog containing Options for hierarchical search.

Input / output data

Input	Image. List of regions of interest. Histogram with threshold (This histogram object is only taken into account if never a threshold has been calculated for the input list of regions).
Output	New list of regions of interest, corresponding to the detected objects. A new data object is created in the data pool.

2.7.4.4.2 Create ROIs by Thresholding: Parameter Dialog

This section describes the settings to be made in the parameter dialog of function **Create ROIs by thresholding**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Create ROIs by thresholding** is selected.

Aside from the parameters set here, the object search is influenced by the threshold computed by check function **Determine threshold**. An important factor is the **required contrast**. If the difference between minimal and maximal gray level inside a region of interest is smaller than this value, no object is detected, even if the thresholding would have separated an object from the background. This helps to avoid that a slight disturbance is accepted as an object, which in fact is missing.

The dialog has the following elements:

Element	Description
Toolbar Image panel	Detailed information on page "Standard parameters of check functions".
Hierarchical search	Opens the dialog box **hierarchical search options** to adjust common settings.

Object color

Lets you select, whether NeuroCheck has to search for light objects on a dark background or vice versa

> Note: when using range thresholding, light and dark are logical rather than visual terms; see check function **Determine threshold** for details.

Object creation

Sets the clipping of found objects:

- **Complete:**
  From every object touched by a search ray a new region of interest is created comprising the full extent of the object.
- **Clipped:**
  Only the part of the object lying inside of the original region of interest (resp. inside the surrounding area of polylines) is used as the new region of interest.
- **Edge only:**
  Only the first edge of the object touched by the search ray is created as new region of interest; again clipped at the borders of the original region. This setting is ineffectual for filled regions of interest, because they do not have a preferred direction. The point where polyline and contour intersect is determined with subpixel precision using an interpolation algorithm. Every measurement created by check function **Gauge ROIs** that refers to such an edge will therefore be computed with subpixel precision.

Fill regions

If this check box is activated the contour of the found objects is completely filled, including areas with the opposite color, otherwise these areas are excluded from the object. This is important for area calculations.
Example: the image contains a circular ring. If this object is created with **Fill regions** activated, the object's area will be the area within the outer circle; if **Fill regions** is not activated, the area will consist of the ring only, i.e. an area calculation using function **Compute Features** will yield the area of the outer circle minus the area of the hole.

Advanced…

Opens the dialog box **Advanced Options** to adjust settings of the search algorithm.

Object selection

Selects, which objects are used:

- **All objects:**
  All detected objects are used.
- **First object:**
  Only the object detected first is used.
- **Largest object:**
  Only the largest detected object is used.

2.7.4.5 Template Matching

2.7.4.5.1 Template Matching: Introduction

Function

This check function creates new regions of interest starting from existing regions of interest by a template search process. From the detected objects a new list of regions of interest is created, that can be processed further. All regions of interest created from within regions belonging to the same group will form a group with the same number in the new list of regions. The templates to be searched for are created interactively using the **Template Editor**.

Details of the procedure

- The main advantage of this method compared with **Create ROIs by thresholding** is that the check function can find non-contiguous objects as a single region (e.g. an "i" together with its dot or the typical characters created by needle stamping machines). Also it can be used for character recognition directly because of the class information it generates, sufficient quality of the patterns provided.
- Disadvantages are the necessity for a-priori knowledge of the objects to be found and the amount of computation required for the correlation (though there are several optimization methods in NeuroCheck). On page "Optimization of execution time" you'll find further information to reduce computation time.
- The check function generates the *n* objects as new regions of interest which exhibit the highest degree of correspondence with one of the templates. The number *n* can be set at will (optionally per group).
- The degree of similarity between object and template is determined from the correlation coefficient.
- The check function performs a pyramidal search, starting with steps of several pixels, reducing this to a single pixel in the neighborhood of promising positions to speed up computation.

Properties

Check function group Objects

The check function generates feature information. See section "Template features" for a list of features computed by this check function.

The check function generates class information which can be evaluated using function **Evaluate Classes**. Classes are edited using the **Template Editor**.

The check function has parameters that can be changed dynamically.

The check function can output result data. It sends resp. writes the number of created ROIs.

The check function has a Parameter dialog.

Input / output data

Input	Image. List of regions of interest.
Output	New list of regions of interest, corresponding to the detected objects. A new data object is created in the data pool. New image displaying various results. A new data object is created in the data pool.

2.7.4.5.2 Template Matching: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Template Matching**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function is selected.

The dialog has the following elements:

Element	Description
Toolbar, Image panel, Group parameters	Detailed information on page "Standard parameters of check functions".
Templates...	This button opens the Template Editor for creating and editing templates. The info box to the right displays the number of actually used classes.
Result image	Select from this list the contents to be displayed in the output image of the function. See Template Output Images for details.
Classes	**All**: If this option is selected, the function will search for templates of all available classes. This is useful if there is no a-priori information about the location of specific types of objects in the image. **Single class**: If this option is selected, the function will search only for templates of the class selected in the adjacent list box within the available regions (or the selected group of regions). If it is known, where objects of a specific type will be located in specific image areas this will speed up the search, because the function will have to perform a smaller number of correlation operations.
Rotate templates	If this checkbox is selected, additional templates are generated internally to match rotated patterns in the image. For each angle increment within the given angle range, a template is generated and used in the matching process.
Angle range	Sets the range around the reference orientation of 0° within which additional templates will be generated.
Increment	Sets the difference of the angle between generated templates within the given range.

Minimum quality	Required degree of correspondence (between 1 and 100%) to the template an object must reach to be accepted as new region of interest. The degree of correspondence equals the correlation coefficient of object and template. On color images the minimum quality is corresponding to the arithmetical mean of the three quality values of color channels.
Result positions	In this box you can set the number of objects the function will create at most inside the available regions (or the selected group of regions). Positions are sorted according to matching quality, i.e. correlation coefficient. If this value is set to one only the best matching object will be created.
Advanced...	This button opens the dialog box **Template Matching Options** to adjust settings of the search algorithm.

2.7.4.5.3 Template Matching Features (Classification result)

This section describes the features computed by function **Template Matching**. The features can be used by functions like **Sort ROIs**, **Screen ROIs**, and **Classify ROIs**.

The following features are computed and added to the features of every region of interest found by this function:

Feature	Description
ID No.	Numerical order of detected regions of interest (not the group number) starting with the highest class quality.
Class	Index of the class to which the template belongs, which exhibits the greatest similarity to the region. Template classes are edited on page **Template Classes** of the **Template Editor**.
Class quality	Correlation coefficient between the region and the template, which exhibits the greatest similarity to the region.
X Position	X coordinate of the top left corner of the newly created region of interest.
Y Position	Y coordinate of the top left corner of the newly created region of interest.
Rotation angle	Angle of rotation under which the pattern has been found; computed only if the **Rotate templates** option has been activated in the parameter dialog of the check function.
Group No.	The Group number is corresponding to the groups of ROIs defined in check function **Define ROIs**. The group number is only available if **group parameters** was selected in the parameter dialog. In the other case this column use standard value "0".

2.7.4.5.4 Template Editor

2.7.4.5.4.1 Template Editor

The **Template-Editor** is opened by choosing "**Templates**…" in the parameter dialog of check function Template Matching. Here you can choose between adding new classes and templates or editing existing templates, if there are already templates defined. Otherwise you will have to add templates first. This area lists the classes currently defined in the parameter set. You can edit the name of a class in place by clicking it two times with the left mouse button (not a double-click!). You can resize the dialog window via mouse on the window frame.

The page contains the following elements:

Element	Description
Tree view	**Structure:** 1st level: Classes 2nd level: Templates **Handling:** Organize the tree content via four next buttons.
Add Class…	Here you can specify the different classes which the patterns in your template set belong to. Template sets should not contain empty classes, so you should not specify classes for which there are no patterns available on your test pieces. Of course you can add and delete classes later at will. A template must belong to a class, so you must specify at least one class.
Add template…	Adds new template using a dialog. Further information on page "Create Template".
Import Template…	Imports *.tif files using windows standard dialog. NeuroCheck saves template images and its correlation points in *.tif- files e.g. to use them in other check routines too.
Delete!	Deletes selected element. **Classes:** This button will delete the currently selected class from the template set. Note that this will also delete all templates belonging to this class. **Templates:** Deletes the current template.
	Moves the selected template(s) in the next upper or lower class.
Template overview…	It allows you to view all existing templates by class. Selecting one of the templates causes NeuroCheck to jump to the selected template for editing this particular template.
Class information	Only available on class level. Shows the number of templates in the selected class as well as the last modification date and time.

2.7.4.5.5 Optimization of execution time

The search method in the **Template Matching** check function requires time intensive computation. Depending on the parameter dialog settings, run time can be slowed down too much to implement your application.

Take the following steps to accelerate the check function's run time:

- **Reduce the number of templates:**
  Only add templates that are really different. Templates without any hits can be deleted after a thorough review.
- **Reduce the number of indicator points:**
  More often than not you can achieve the same results using 50 or 100 points instead of 200 or 400.
- **"Coarse search step size":**
  Provided the structure size of the desired image object is sufficient, this value can be greater than 4.
- **Option "Angle range" in "Rotate templates":**
  If you have prior knowledge about the orientation of the desired image object, reduce the range as much as possible.
- **Step size**:
  Provided that unambiguity is assured, this value can be increased slightly.

Important note:
Applying any of the optimization measures listed above can impair the reliability and quality of the search result!
After optimization, take several sample parts to ascertain that the results are still reliable and correct.

2.7.4.6 Color Matching

2.7.4.6.1 Color Matching: Introduction

Function

This check function analyzes an image based on color information. It takes an RGB color image as input (provided by check function **Transfer Image** if the image has been captured by a color camera or has been loaded from a true color bitmap) and creates a gray level image of the same size as its input image. The gray level of every pixel in the result image indicates the similarity of the corresponding pixel in the original image to one or more reference colors. Reference colors are created interactively using the **Reference Color Editor**.

The check function can work in three different modes, which are explained in more detail in section "Color matching modes":

1. **Distance mode**:
   the brightness of a result pixel indicates its similarity to a single reference color.
2. **Ratio mode**:
   the brightness of a result pixel indicates ratio of similarity to two reference colors.
3. **Segmentation mode**:
   the brightness of a result pixel encodes the reference color to which it is most similar. In this mode the function creates a region of interest for every contiguous area of uniform gray level in the result image.

Properties

Check function group Objects

The check function generates feature information. See section "Color features" for a list of features computed by this function.

In segmentation mode the check function generates class information which can be evaluated using check function **Evaluate Classes**. Classes are edited using the Reference Color Editor.

The check function has parameters that can be changed dynamically.

The check function can output result data. It sends resp. writes the number of created ROIs.

The check function has a Parameter dialog.

Input / output data

Input	Color image.
	List of regions of interest.

Output	Gray level image. A new data object is created in the data pool. List of regions of interest (empty, except in segmentation mode). A new data object is created in the data pool.

2.7.4.6.2 Color Matching: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Color Matching**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when this check function is selected.

The function has the following parameters:

Parameter	Description
Toolbar, Image panel, Group Parameters	Detailed information on page "Standard parameters of check functions".
Reference colors...	This button opens the **Reference Color Editor** for creating and editing Colors.
Mode	You can select the function's mode of operation from this list box. • **Distance:** Computes for each pixel in the source image the similarity of its color to the reference color. This mode uses only one reference color. • **Ratio:** Computes for each pixel in the source image the difference of its color to both reference colors. This mode uses two reference colors. • **Segmentation:** A distinct gray level is assigned to each reference color class. This mode uses all reference colors. In detail the modes are explained in section "Color matching modes".
Selectivity	In this edit box you can set the minimum required similarity to the reference color an image pixel has to achieve. Pixels with similarity scores below this value are set to black all modes and will not form regions of interest in segmentation mode. The value can be set between 0 and 250, corresponding to similarity scores between 0.0 and 1.0.
Ignore Brightness	If this check box is activated, the function will disregard brightness information from the colors, meaning that it will only regard the ratios of the red, blue and green portions, not their absolute values. With this option, White is exactly the same as Gray or Black, namely a color with identical shares of red, blue and green.
Suppress dark areas	Only available, if checkbox **Ignore Brightness** is activated. This threshold prevents that subdued results were wrongly "found" on dark areas.

Color 1	In distance mode, this list box sets the color to which the individual points in the image are to be compared. In ratio mode it selects the primary color for the ratio computation.
Color 2	In ratio mode this list box selects the secondary color for the ratio computation
Options	Limitation of the minimum and maximum size of created objects. Reduces the among of founded objects. (e.g. to exclude disturbing midget areas or pixel.)

2.7.4.6.3 Color Matching Features

This section describes the features computed by function **Color Matching**. The features can be used by functions like **Sort ROIs**, **Screen ROIs**, and **Evaluate Classes**.

The following features are computed and added to the features of every region of interest created by this function in segmentation mode:

Feature	Description
Class	Index of the class to which the color belongs, which exhibits the greatest similarity to the region. Color classes are edited using the Reference Color Editor.
Class quality	Similarity score between the region and the color, which exhibits the greatest similarity to the region.

See section "Color matching modes" for an explanation of the segmentation mode.

2.7.4.6.4 Reference Color Editor

2.7.4.6.4.1 Reference Color Editor: Introduction

The **Reference Color Editor** is opened by choosing **Reference colors…** in the parameter dialog of check function Color Matching. In the dialog you can:

- add classes and reference colors
- edit or remove existing classes and reference colors.

The page contains the following elements:

Element	**Description**
Tree view	**Structure:** 1. Level: Classes 2. Level: Reference colors You can organize the content of the tree via the lower buttons.
Add Class...	On this page you can specify the different classes which your reference colors belong to. Color sets should not contain empty classes, so you should not specify classes for which there are no colors available on your test pieces. Of course you can add and delete classes later at will. A color must belong to a class, so you have to specify at least one class.
Add reference color...	Adds new reference color using a dialog. Find more information in section "Add reference color".
Delete!	Deletes the selected element: **Classes:** The button will delete the currently selected class from the color set. Note that this will also delete all reference colors belonging to this class. **Reference colors:** Deletes the current color.
Adjust selected color...	Opens the dialog **Adjust reference color** to edit the selected color.
	Moves selected reference color to next lower or upper class.

2.7.4.6.5 Color Matching Modes

2.7.4.6.5.1 Color Similarity

This section explains the color similarity concept and the different operating modes of check function **Color Matching**.

Color similarity

A color is represented in NeuroCheck by three values for the three basic colors, Red, Green and Blue. The values fall into the standard gray level range from 0 to 255.

The similarity between two colors is expressed as the 1 minus the Euclidean distance between the colors, normalized to 1:

$$s(c1, c2) = 1 - \frac{\sqrt{(r1 - r2)^2 + (g1 - g2)^2 + (b1 - b2)^2}}{\sqrt{3 * 255^2}}$$

with `c1, c2` as the two colors, `rN, gN, bN` as their respective red, green and blue values.

Note that the selectivity setting of function **Color Matching** maps this similarity value to a range from 0 to 250.

The representations for some simple colors and their similarity to the pure bright colors:

Color	Red	Green	Blue	s(red)	s(green)	s(blue)
White	255	255	255	0.184	0.184	0.184
Bright red	255	0	0	1.0	0.184	0.184
Bright green	0	255	0	0.184	1.0	0.184
Bright blue	0	0	255	0.184	0.184	1.0
Medium red	128	0	0	0.71	0.354	0.354
Medium green	0	128	0	0.354	0.71	0.354
Medium blue	0	0	128	0.354	0.354	0.71
Medium gray	128	128	128	0.499	0.499	0.499

Note that Medium gray is much more similar to the reference colors than pure White. This is due to the difference in overall brightness. The brightness of medium gray is much closer to the brightness of a pure basic color than to that of pure white. This can be changed by using "hue only processing" explained in the section "Hue only processing".

2.7.4.6.5.2 Hue Only Processing

The hue of a color is defined as the direction of the color vector, without regard to its absolute length, which gives the brightness of the color. The **Ignore brightness** option of check function **Color Matching** works by normalizing the color vectors in such a way that the maximum component is always 255. Two color vectors sharing the same hue will then be identical. The only colors in the table in section "Color Similarity" which will be changed by this process are those of medium brightness. Note that medium gray now shows exactly the same behavior as white and medium green the same as pure green etc.:

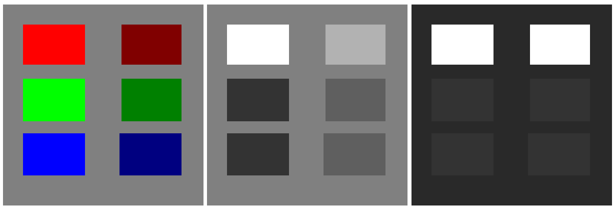
Color	Red	Green	Blue	s(red)	s(green)	s(blue)
White	255	255	255	0.184	0.184	0.184
Bright red	255	0	0	1.0	0.184	0.184
Bright green	0	255	0	0.184	1.0	0.184
Bright blue	0	0	255	0.184	0.184	1.0
Medium red	255	0	0	1.0	0.184	0.184
Medium green	0	255	0	0.184	1.0	0.184
Medium blue	0	0	255	0.184	0.184	1.0
Medium gray	255	255	255	0.184	0.184	0.184

2.7.4.6.5.3 Color Distance Mode

In distance mode, a single reference color is used. For each pixel in the source image, the similarity of its color to the reference color is computed. The gray level of the corresponding pixel in the result image indicates the degree of similarity. A similarity below the required selectivity value is coded as black, a similarity of 1.0, which means that the pixel has exactly the reference color, is coded as white. The example image contains on the left three areas of pure color with full brightness, on the right three areas of pure color with medium brightness.

Using reference color red, the output will be as in the following images:

Example image **Absolute** **Hue only**



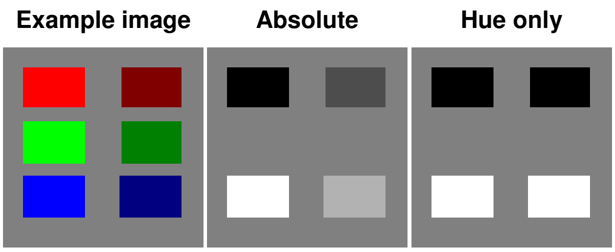
Since all the colors are pure colors, except for the background, the similarity scores are identical for the blue and green areas. Note that in "Hue only" mode the medium red is treated identically to the bright red, whereas the background becomes much darker, because it is treated as white and so loses the similarity to the pure color in overall brightness, as explained in section "Color Similarity".

2.7.4.6.5.4 Color Ratio Mode

In ratio mode, two reference colors are used. For each pixel in the source image, the difference of its color to both reference colors is computed. The gray level of the corresponding pixel in the result image indicates the relationship between the difference values in the following way:

$$g = \max\left(\min\left(\frac{128 * (d2 - d1)}{(d2 + d1)} + 128, 252\right), 3\right)$$

The effect is that a pixel having exactly the first reference color (`d1=0`) is set to almost White in the result image (the computation yields 256, which is clipped to 252 by the `min()` function above). A pixel having exactly the second reference color (`d2=0`) is set to almost Black (the computation yields 0, which is set to 3 by the `max()` function). Other pixels receive intermediate gray levels. Pixels with both distances outside the required selectivity range (`1-d1 < s AND 1-d2 < s`) are set to absolute Black (0). Setting these colors to a black background retains the meaning of the selectivity in distance mode as a color range not to be considered. Using pure red and pure blue as reference colors, the output for the example image will be as follows (for a selectivity of 0):



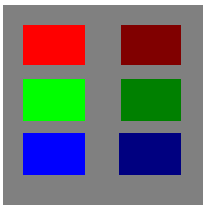
The brightness of the areas indicates their location with respect to both reference colors. Note that the originally green areas appear exactly like the background, because their distances from red and blue are identical. In **Ignore brightness** mode, the distinction between bright and medium colors vanishes. With increasing selectivity, first the green areas, then the gray background will be mapped to absolute black, because their distance to both reference colors becomes too large.

> Note that blue is mapped to almost white, red to almost black, because matching is done in alphabetical sequence of class names.

2.7.4.6.5.5 Color Segmentation Mode

In segmentation mode, all reference colors are used. A distinct gray level is assigned to each reference color class. Every pixel is painted in the gray level belonging to the reference color class to which it is most similar, provided that this similarity is larger than the selectivity value. The following examples use an image with six areas of pure colors, three of full brightness and three of medium brightness. The reference colors are pure red, pure green and pure blue, all with full brightness.

Example image



2.7.4.7 Create Edges

2.7.4.7.1 Create Edges: Introduction

Function

This check function creates new regions of interest based on an edge model. According to this model, an edge is defined as a gray level transition of at least **height** gray levels over at most **length** pixels, as explained in section "Edge model". For color images the check function uses an arithmetic mean which is based on the calculated gray levels of each color channel. This procedure ensures the edge detection in all three channels.

The check function works along lines, polylines or, within a rectangular region of interest, along search rays parallel to the borders of the rectangle. Wherever it finds a change in brightness that corresponds to the model it creates a portion of the edge that is 10 pixels long, regardless of the surrounding area of the polyline. The center point of that edge section is computed with subpixel precision. All measurements referring to that point are therefore subpixel precise. The check function has been designed especially for gauging applications as an alternative to check function **Create ROIs by Thresholding**.

It has the following advantages for this purpose compared to that function:

- Less sensitivity to absolute brightness level; the edge is defined in terms of the gray level difference, not as an absolute threshold.
- Less sensitivity to surface texturing, as shown in the example below.
- Faster, because it does not have to check the complete object contour.

The function can use a global edge model or separate edge models for each group of regions. From the detected edge sections a new list of regions of interest is created, that can be processed further. All regions of interest created from within regions belonging to the same group will form a group with the same number in the resulting list of regions.

Properties

Check function group Objects

The check function has parameters that can be changed dynamically.

The check function has a Parameter dialog.

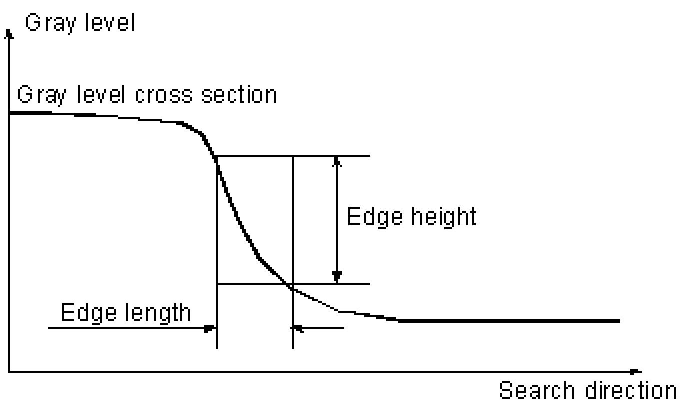
Input / output data

Input	Image. List of regions of interest.
Output	New list of regions of interest, corresponding to the detected edges. A new data object is created in the data pool.

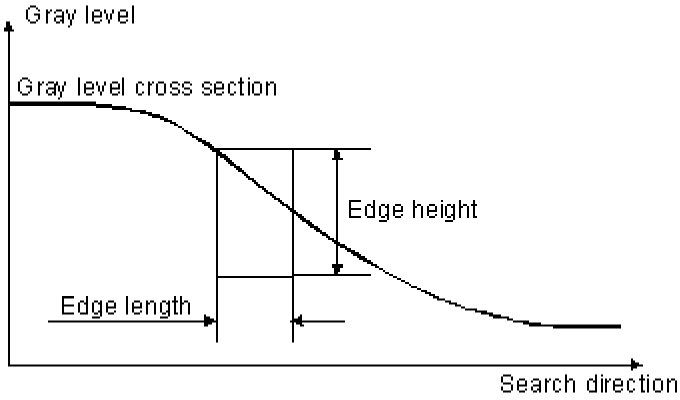
2.7.4.7.2 Edge Model

This section describes the edge model used by function **Create Edges** in more detail.

The model defines an edge as a gray level transition of at least **height** gray levels over at most **length** pixels, as shown in the following image.



If the points, between which gray level difference of the required **height** occurs along the search line, are farther apart than the **length** specifies, the function does not regard the transition as an edge, as shown in the following image:



RGB color images

Each color channel (red, green and blue) will be separately interpreted as gray level value. The check function calculates the gray level difference vector for each channel according to the example above.. The result value is the arithmetical mean of these three difference vectors.

2.7.4.8 Combine ROIs

2.7.4.8.1 Combine Regions of Interest: Introduction

Combines lists of regions of interest generated by different check functions. The first created list in the data pool will be appended to the second list. The second list is the result list.

Properties

Check function group Objects

The function has a parameter dialog:

Element	Description
Offset of group numbers	Value by which the group numbers of the first list of regions of interest will be increased. If smaller than the highest group number in the second list, regions of interest from the first list will be added to groups of the second list. From the highest group number onwards individual group assignments will be retained, higher values enable later addition of groups to the second list of regions of interest.

Input / output data

Input	Image (for display only).

	List of regions of interest 1, will not be changed. List of regions of interest 2, will be enlarged.
Output	Enlarged list of regions of interest, corresponding to input list 2 in the data pool. No new data object is created in the data pool.

2.7.4.8.2 Example for group offsets

Assume two lists of regions. Each contains two groups, numbered 0 and 1 respectively. The following configurations of the result list are possible:

- Offset 0: regions from identically numbered groups are merged into the same result group.
- Offset 1: regions from group 0 in the first source list are put into group 1 of the result list (together with the objects from group 1 in the second source list) because of the group number offset. An additional group 2 is created in the result list for the regions from group 1 in the first source list, whereas result group 0 will contain only objects from the second source list.
- Offset 2: in this case all groups become separated. The result list will contain two new groups, number 2 and 3, with the regions from groups 0 and 1 of the first source list.

The following table summarizes the settings:

Offset	**Source list 1**	**Source list 2**	**Result list**
0	group 0: 3 objects group 1: 5 objects	group 0: 4 objects group 1: 6 objects	group 0: 7 objects group 1: 11 objects
1	group 0: 3 objects -> group 1 group 1: 5 objects -> group 2	group 0: 4 objects group 1: 6 objects	group 0: 4 objects group 1: 9 objects group 2: 5 objects
2	group 0: 3 objects -> group 2 group 1: 5 objects -> group 3	group 0: 4 objects group 1: 6 objects	group 0: 4 objects group 1: 6 objects group 2: 3 objects group 3: 5 objects

2.7.4.9 Copy ROIs

2.7.4.9.1 Copy ROIs: Introduction

Function

This check function generates an exact copy of an existing list of regions of interest. This is useful for applying several different processing sequences to a list of regions, because functions like **Screen ROIs** delete regions completely, making them unavailable for further processing.

Properties

Check function group Objects

The function can output result data.

The function does not have parameters.

Input / output data

Input	Image (for display only). List of regions of interest.
Output	Copied list of regions of interest. A new data object is created in the data pool.

2.7.4.10 Split ROIs

2.7.4.10.1 Split Regions of Interest: Introduction

Splits regions of interest in its input list at vertices, i.e. points of high curvature on their contour.

Properties

Check function group Objects

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Creation	**Corners:** If this option is selected, parts of the contour around the points of highest curvature are created as new objects. In effect this removes the smooth parts of the contour and leaves only the corners.

	Lines: If this option is selected, parts of the contour between the points of highest curvature are created as new objects. In effect this removes the corners from the contour and leaves only the smooth parts.
Corner type	**All (Corners):** If this option is selected, all corners are used, convex and concave. **Convex:** If this option is selected, only the convex corners are used, i.e. those pointing outward. **Concave:** If this option is selected, only the concave corners are used, i.e. those pointing inward.
Calculation Parameters	**Number of expected corners:** Enter here the nominal number of corners. If this number is denoted as *N*, the function will use the *N* points of highest curvature on the contour. **Corner range:** Enter here the number of points at each side of the corner which are: - included, when corners are created - removed, when lines are created **Smoothing:** Enter here the length of the chord to be used for measuring the curvature. You will find more information on measuring curvature in section **Compute Curvature**.

Input / output data

Input	Image. List of regions of interest.
Output	List of regions of interest. A new data object is created in the data pool.

2.7.4.11 Compute Model Geometries

2.7.4.11.1 Compute Model Geometries: Introduction

Function

Computes model geometries, like ideal circles or straight lines from existing regions of interest. This computation has some notable properties:

- The check function can compute model geometries either for individual regions of interest or for whole groups of regions. It can for example approximate the edge of an object by a straight line or compute a regression line from the center points of several objects in order to estimate how well objects are aligned.
- Computing a model geometry for a single object replaces the object’s center of gravity with the center point of the model geometry. Because of the averaging process this point is subpixel precise (with the exception of model lines computed from edges precisely parallel to the X- or Y-axis. In this case no information for the averaging process is available perpendicular to the direction of the edge). For more information about the relationship between model geometries and subpixel precision gauging please refer to section "Subpixel precision gauging".
- Computing a model geometry for a group of objects has the additional effect of replacing the individual objects in the list of regions of interest with a compound object consisting of the reference points for these objects and the lines connecting these points. That means, if you compute a straight line from a row of objects, you will not be able to access those objects later, but instead it will be possible to use the computed line immediately , e.g. to search for other objects along this line.

Properties

Check function group Objects

The check function has a Parameter dialog.

Input / output data

Input	Image. List of regions of interest.
Output	List of regions with computed model geometries; the list may be changed as explained above. No new data object is created in the data pool.

2.7.4.11.2 Line Computation

The line generation algorithm of function **Compute model geometries** works as follows:

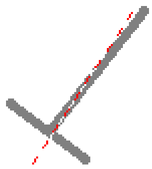
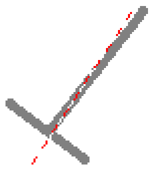
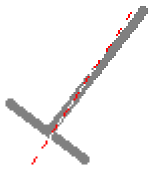
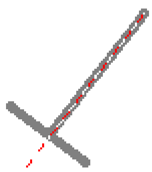
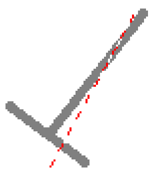
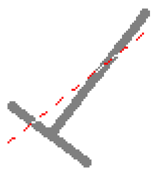
1. A standard regression line is computed from all points of the object contour (or the reference points of all objects in the group).
2. Corresponding to the number of iterations set in the parameter dialog additional steps are performed consisting of:
    - eliminating points on one or both sides of the regression line,
    - recomputing the regression line with the remaining support points.

The number of support points to retain is determined by the **Support points percentage** slider in the parameter dialog. A low setting of this slider effectively retains only points very near to the current regression line, whereas a high setting also retains points farther away. The location of support points to be eliminated is determined by the **Creation** setting in the parameter dialog:

- **Center line** eliminate points on both sides of the regression line, thus reducing the effect of "outliers" and moving the regression line closer to the perceived principal direction of the part.
- **Right line** eliminate points on the left side of the regression line, thus moving the regression line towards the right border of the object contour.
- **Left line** eliminate points on the right side of the regression line, thus moving the regression line towards the left border of the object contour.

The following table shows the effect of the algorithm for 1 and 10 iterations.

Please note that the test piece is not perfectly symmetrical, the left arm being slightly longer than the right, so the center line does not run straight along the perceived axis in the first iteration.

	Center line	**Right line**	**Left line**
1 iteration			
10 iterations			

The strength of the effect also depends on the percentage of support points retained, so with a low percentage of support points, less iterations are necessary and the line can be shifted further outwards, but too small a percentage may lead to surprising effects.

When the line is computed for a whole group of objects, the effect is similar, but instead of points on the contour of one object, the reference points (usually the centers of gravity) of all objects in the group are used as support points.

2.7.4.11.3 Circle Computation

The circle generation algorithm of function **Compute model geometries** works as follows:

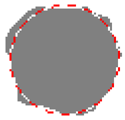
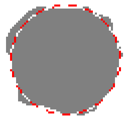
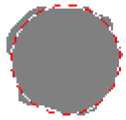
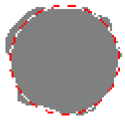
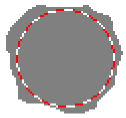
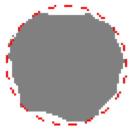
1. A standard approximation circle is computed from all points of the object contour.
2. Corresponding to the number of iterations set in the parameter dialog additional steps are performed consisting of:
    - eliminating points on one or both sides of the approximation circle,
    - recomputing the circle with the remaining support points.

The number of support points to retain is determined by the **Support points percentage** slider in the parameter dialog. A low setting of this slider effectively retains only points very near to the current circle, whereas a high setting also retains points farther away. The location of support points to be eliminated is determined by the **Creation** setting in the parameter dialog:

- **Medium circle**:
  eliminates points on both sides of the circle, thus reducing the effect of "outliers" and moving the circle closer to the perceived average contour of the part, but the difference may not be easy to see on the nearly circular objects on which this function will normally be used.
- **Minimal circle**:
  eliminates points outside of the circle, thus reducing the diameter down to an inner circle, lying completely within the contour.
- **Maximal circle**:
  eliminates points within the circle, thus enlarging the diameter up to an outer circle, completely enclosing the contour.

The following table shows the effect of the algorithm for 1 and 10 iterations.

The part being near circular, the effect of the iterations on the medium circle is hard to perceive.

	Medium circle	Minimum circle	Maximum circle
1 iteration			
10 iterations			

The strength of the effect also depends on the percentage of support points retained, so with a low number of support points, less iterations are necessary and the circle can be shifted further outwards or inwards, but too small a number of points may lead to surprising effects.

When the circle is computed for a whole group of objects, the effect is similar, but instead of points on the contour of one object, the reference points (usually the centers of gravity) of all objects in the group are used as support points.

Computing a maximum or minimum circle for a group of objects representing only part of the circle may lead to visually surprising results, because the circle center is optimized together with the radius. Removing points within the circle may therefore lead not to a widening of the circle, but instead yield a smaller circle with a different center point.

2.7.5 Analysis

2.7.5.1 Introduction to Check Function Group "Analysis"

The check functions in group **Analysis** are used for analyzing objects in the image. In contrast to the check functions of group **Preprocessing** they always require at least one region of interest to work with. Manual definition of a region of interest is therefore a prerequisite for using any of these check functions.

Some check functions serve the purpose to calculate object features. Other check functions in this group decide about the validity of objects or form the basis of decisions about faults on a test piece.

2.7.5.2 Compute Features

2.7.5.2.1 Compute Features: Introduction

Function

This function computes various scalar values for every region of interest in its input list. The measurements belong to the features in NeuroCheck. The function computes the same set of measurements for every region of interest in its input list. Features can be computed for manually defined regions of interest, e.g. to determine the average brightness of a specified image region, but mostly this function will be used for automatically created regions of interest, i.e. the results of an object search.

Geometrical features (e.g. dimensions, coordinates etc.) are computed according to the conversion factor determined by one of the function **Calibrate pixels** or **Calibrate measurements**. If no calibration has taken place within the check, NeuroCheck assumes a size of 1mm for each image pixel.

For a list of all features this function can compute refer to section "Feature list".

Display of the feature list

After the check function has been carried out, a row is created for each ROI. All ROIs are treated separately (i.e., existing ROI groups are ignored). The result image shows the ROIs with consecutive numbering from the generated feature list.

There is a column with results for each feature that has been selected in the parameter dialog. In each column, the minimum and maximum value is indicated.

The column heads correspond to the feature names from the parameter dialog. You can name features individually to have better control about the value measured by your inspection system. Please refer to section "Compute features: parameter dialog" for details.

Properties

Check function group Analysis

The function generates feature information. See section "Feature list" for a list of features computed by this function.

The function can output result data. It sends resp. writes the computed values.

The function has a Parameter dialog.

Input / output data

Input	Image. List of regions of interest.
Output	List of regions with computed feature values. No new data object is created in the data pool.

2.7.5.2.2 Compute Features: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Compute features**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Compute features** is selected.

The dialog has the following elements:

Element	Description
Categories	Double click to open via left mouse button or via icon "+". Features are subdivided into the following sections: **Basic geometry:** Contains frequently used features (e.g. area, perimeter, etc.). **Advanced geometry:** Contains special and rarely used features (e.g. number of holes, radii, axis features, etc.) **Gray level:** Contains features of gray level statistics and gradients. **Miscellaneous:** Contains features that do not fit into one of the other categories.
Features	The features to be computed are selected from this list by clicking the check box in front of the feature name. A second click into the box deselects the feature. You can also rename the features in-place by clicking a selected feature a second time and editing the name. Clicking the list with the right mouse button opens a context

	menu allowing you to reset all edited feature names to the original designations.
Help	This button provides additional information about the feature currently highlighted in the feature list.

2.7.5.2.3 Feature List

2.7.5.2.3.1 Feature List

The features which can be computed by function Compute features are divided into four categories:

1. Standard Features
2. Advanced Geometry Features
3. Gray Level Features
4. Miscellaneous Features

Standard Features

- Area
- Perimeter
- Form factor
- Origin X
- Origin Y
- Center of gravity X
- Center of gravity Y
- White pixels
- Black pixels

Advanced Geometry Features

- Size X
- Size Y
- Ratio
- Lower right X
- Lower right Y
- Border contact general
- Border contact top

- Border contact bottom
- Border contact left
- Border contact right
- Number of holes
- Radius, average
- Radius, minimal
- Radius, maximal
- Radii, angle
- Axis, length major
- Axis, length minor
- Axis, orientation [0…180]
- Axis, orientation [0…360]
- Fibre length
- Fibre width
- Elongation
- Minimum Feret Diameter
- Maximum Feret Diameter

Gray Level Features

- Gray level, average
- Gray level, minimal
- Gray level, maximal
- Gray level, standard deviation (std. dev.)
- Contrast
- Gradient, average
- Gradient, maximal
- Gradient, standard deviation (std. dev.)

Miscellaneous Features

- Group number

2.7.5.3 Screen ROIs

2.7.5.3.1 Screen Regions of Interest: Introduction

Function

After executing this function, a list of regions of interest consists only of those regions whose features meet all criteria set within this function. The function checks for each activated feature if the feature value of the object meets the range defined by a minimum and a maximum.

Four modes for application of the range values are available:

- Regions with feature values inside the range given by minimum and maximum are valid (usual use case).
- Regions with feature values less than the maximum are valid.
- Regions with feature values greater than the minimum are valid.
- Regions are treated as valid, whose feature value is either less than the minimum or greater than the maximum. Here all objects will be deleted, whose features are inside the defined range.

Regions whose feature value is identical to the given minimum or maximum value are accepted as valid. This is especially important for the value 0. Therefore you should not use it as a limit value. Also please take care of the fact that the feature values are displayed using five decimal places, whilst their actual value may differ slightly from this.

Properties

Check function group Analysis

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

- The check function has an additional dialog field, which displays the Feature Value Distribution (Statistics).

Input / output data

Input	Image. List of regions of interest with feature values.
Output	Modifies the input list or ROIs. No new data object is created in the data pool.

2.7.5.4 Sort ROIs

2.7.5.4.1 Sort Regions of Interest: Introduction

Function

This function sorts the regions of interest in its input list according to scalar measurements created previously by other functions in group Object features.

Sorting can be done in three modes:

1. **Globally**:
   in this case there is one set of sorting parameters and the complete list of regions will be sorted as a whole.
2. **Per group with individual parameters**:
   in this case, individual sorting criteria can be set for each group of regions and each group will be sorted separately according to its parameters.
3. **Per group with global parameters**:
   in this case, each group will be sorted separately, but with the global parameter set. This mode is useful, for example, in character recognition, when three lines of text are to be sorted in x direction. The sorting criteria are identical for each group, but still each group (i.e. line of text) has to be sorted individually.

Properties

Check function group Analysis

The function has a Parameter dialog.

Input / output data

Input	Image. List of regions of interest.
Output	Modifies the current list of ROIs. No new data object is created in the data pool.

2.7.5.5 Resample ROIs

2.7.5.5.1 Resample Regions of Interest: Introduction

Function

This function converts the contents of the enclosing rectangle of each region of interest in its input list to a resampled image. Resampled images belong to the features used by NeuroCheck to describe regions of interest. They are required for direct classification of visual information.

The original content of the region of interest is subdivided into fields of uniform size for resampling. Each field is represented by one point in the resampled image.

If the input image is a color image, the resampled images will be gray level images though.

Properties

Check function group Analysis

The function generates feature information. It adds a normalized image to the features of all regions of interest in its input list.

The function has a parameter dialog.

The parameter dialog consists of the following elements:

Element	Description
Result size	Sets the number of pixels in each resampled image in horizontal and vertical direction.
Resampling mode	Sets the method used for computing the brightness of the pixels in the resampled image: • **Average gray level:** the brightness of the original image field is averaged (better representation of the original image content, but slower). • **Central gray level:** the brightness of the central point in the original field is used.
Normalize contrast	If this check box is activated, the gray levels inside the resampled image section are converted to span the full gray level range. Normalization will be skipped if the input region has very low contrast to avoid the amplification of noise.

Input / output data

Input	Image. List of regions of interest.

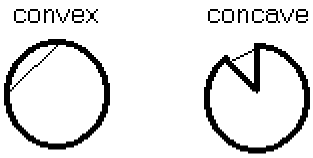
Output	List of regions with attached resampled images. No new data object is created in the data pool.

2.7.5.6 Compute Curvature

2.7.5.6.1 Compute Curvature: Introduction

Function

This check function computes several values describing the curvature of the contours of all regions of interest in its input list. The curvature is evaluated by connecting pairs of points on the contour with a straight line and computing the distance of the contour from the line, as indicated in the following image:



Parts of the contour where the chord lies to the right of the contour (with respect to a clockwise movement on the contour) are called convex, others concave. The computation can be restricted to concave parts, which is useful, for example, if you are looking for dents in the contour of an otherwise convex object as in the above image. The reverse is also possible.

See also:

- **Statistics**

  From the curvature values of all pairs of points on the contour the following statistical values are calculated: minimum, average, maximum curvature, standard deviation and amplitude. You will find details about these statistical values in section "Curvature statistics".

- **Signature**

  The check function can also return a vector containing a curvature value for each point on the contour called a signature. As it can require a lot of space, it is only created upon request. Properties and applications of the curvature vector are explained in more detail in section "Curvature signature".

Properties

Check function group Analysis

Statistical values and signatures can be used as features. See section "Curvature features" for more information.

The check function has a Parameter dialog.

Input / output data

Input	Image. List of regions of interest.
Output	List of regions with computed curvature values and optionally signature. No new data object is created in the data pool.

2.7.5.6.2 Curvature Features

This section describes the features computed by check function **Compute Curvature**. The features can be used by check functions like **Sort ROIs**, **Screen ROIs**, and **Classify ROIs**.

The following features are computed and added to the features of every region of interest in the check function's input list:

Feature	Description
Statistical values	Average, minimum, maximum, standard deviation and amplitude (difference between minimum and maximum) of curvature values along the contour; computed always (see section "Curvature statistics" for details).
Signature	Vector of curvature values, computed only upon request. Currently only check function **Classify ROIs** is able to make use of this vector feature. See section "Curvature signature" for details.

2.7.5.7 Count ROIs

2.7.5.7.1 Count ROIs: Introduction

Function

Counts the number of regions of interest in its input list and compares it to a prescribed range. The range can be set separately for each group of ROIs, if grouping is active.

Properties

Check function group Analysis

The check function is a decision check function.

The check function has target values that can be changed dynamically.

The check function can output result data.

The check function has a target value dialog for setting the valid object counts.

The target value dialog has the following elements:

Element	Description
Toolbar, Image panel, Group parameters	Detailed Information in section "Standard parameters of check functions".
Verify target values	Checkbox activated: The check function acts as a decision check function, compares the existing number of regions with the target values and returns "not O.K." in case of a mismatch. Checkbox deactivated: otherwise the function simply counts the existing regions.
Minimum	To be valid, the group has to contain at least this number of ROIs;
Maximum	To be valid, the group has to contain at most this number of ROIs;

If both values are identical, the group will only be valid if it contains precisely that number of ROIs.

You cannot enter a value for an empty group. If a count of 0 means that the checked piece is OK (e.g. if you are looking for defects like scratches or holes), you will need a defective part to set the group count to 0.

Input / output data

Input	Image, for display only. List of regions of interest to be counted.
Output	None.

2.7.5.8 Print Quality Inspection

2.7.5.8.1 Print Quality Inspection: Introduction

Function

This function creates an image containing the differences between the current image and a reference image inside the regions of interest present in the reference image. This image can then be analyzed further to evaluate the deviations between the current image and the reference image. This approach is typically used for print quality inspection, hence the name of the function. The main problem of print quality inspection is the presence of severe deviations along edges, because the position of character edges in different images is seldom precise to a single pixel. To avoid this problem, the function offers sophisticated hierarchical positioning options, explained in section "Positioning for print quality inspection".

Properties

Check function group Analysis

The function has a Parameter dialog from which several other dialogs are called:

- Options for Global Search
- Options for Local Search
- Reference Image Wizard

Input / output data

Input	Image. List of regions of interest.
Output	Difference image, black outside the regions of interest (no differences are computed outside regions). A new data object is created in the data pool. Copy of the list of regions of interest stored with the reference image. A new data object is created in the data pool.

Note that the input list of regions of interest is used only during teaching of the reference image. During inspection, the function always uses the regions stored together with the reference image. The function can, however, use offset information computed by function **Determine Position** to shift its reference regions relative to the objects contained in the image.

2.7.5.8.2 Print Quality Inspection: Parameter Dialog

This section describes the settings to be made in the parameter dialog of check function **Print Quality Inspection**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Print Quality Inspection** is selected.

The dialog contains the following elements:

Element	Description
Toolbar, Image panel	Detailed Information on page "Standard parameters of check functions".
Position adjustment	If activated, the function uses horizontal and vertical offset information previously computed by function **Determine Position** to shift its reference regions relative to the objects contained in the image.
Brightness compensation	If activated, the function compensates for changes of the average brightness inside the comparison areas. This means that it will not be considered a deviation if the area as a whole becomes uniformly darker or brighter.
Global search	If activated, the function will shift each comparison area within a vicinity of its prescribed position, until it matches the image content optimally. The adjacent **Options** button opens the **Options for Global Search** dialog for setting parameters for this positioning stage.
Local search	If activated, the function will subdivide each comparison area into subareas and shift these within an elasticity range around their original positions, until they match the image content optimally. The adjacent **Options** button opens the **Options for Local Search** dialog for setting parameters for this positioning stage.
Subtraction mode	This group box offers the following options for computing the result image: • **All deviations**: the absolute value of the difference between current image and reference image is used. • **Light**: the result image will contain only bright defects, i.e. pixels where the comparison areas in the current image are brighter than those in the reference image. • **Dark**: the result image will contain only dark defects, i.e. pixels where the comparison areas in the current image are darker than those in the reference image.

	Note that all defects will appear bright in the result image; these options only define, which defects will be regarded at all.
Teach	Opens the **Reference Image Wizard** for storing the reference image.

2.7.5.8.3 Positioning

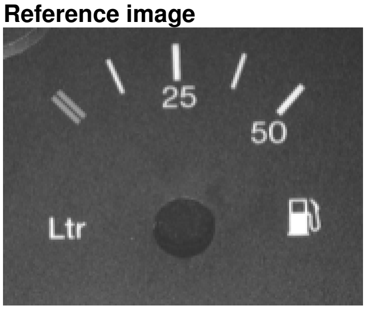
2.7.5.8.3.1 Positioning Challenge

The creation of a difference image for print quality inspection or any other type of reference image comparison is far from trivial. Because of positioning and image capturing inaccuracies, edges are practically never aligned precisely so that a simple subtraction would lead to the detection of severe pseudo errors along the edges. Therefore, check function **Print Quality Inspection** uses a highly configurable, sophisticated three-stage positioning process and also takes heed of the structure size of the objects to be checked.

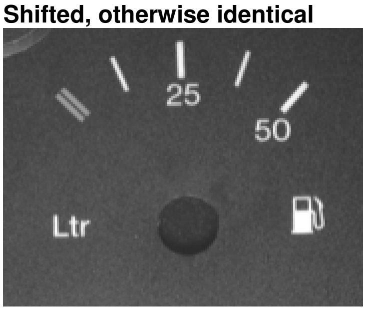
Simple difference image computation

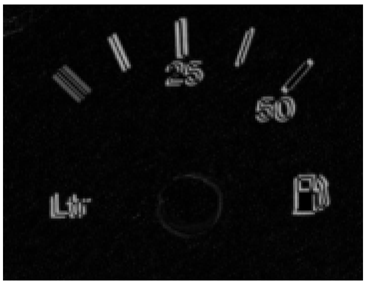
The following images illustrate the above mentioned problem. The second image has been moved very slightly. The human eye will not detect a difference and there is of course nothing wrong with the print, but simply subtracting the images creates notable defects along the edges.

Reference image



Shifted, otherwise identical





Difference image

2.7.5.9 Classify ROIs

2.7.5.9.1 Classify Regions of Interest: Introduction

Function

This function uses a classifier, e.g. a neural network to order regions of interest into classes according to their features.. A typical application is the recognition of digits or letters. The class information can be evaluated using function Evaluate Classes.

The function also serves as the starting point for creating training data and classifiers.

Despite all the assistance given the user by NeuroCheck, classification remains a difficult topic, that requires some background knowledge. A basic introduction to classification and step-wise instructions are given in section "Classification" whereas this section and the following explain the commands and dialogs used in connection with object classification in detail.

Properties

Check function group Analysis

The function can output result data. It sends resp. writes the sequence of identified classes.

The function generates feature information See section "Classification features" for a list of features computed by this function.

The function generates class information which can be evaluated using check function **Evaluate Classes**.

The function has a Parameter Dialog described in the following section with parameters for the identification process and control elements to call the Training Data Wizard and the Classifier Wizard.

Input / output data

Input	Image. List of regions of interest.
Output	New list of regions with attached class assignments (possibly reduced to a single group). A new data object is created in the data pool.

2.7.5.9.2 Classify Regions of Interest: Parameter Dialog

This section describes the settings to be made in the parameter dialog of function **Classify ROIs**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when check function **Classify ROIs** is selected.

The dialog contains the following elements:

Element	Description
All regions	If this button is selected, the check function will classify all regions in its input list.
Regions in group no.	If this button is selected, the check function will classify only regions in a single group. The group is selected from the adjacent list box or by clicking the **Select** button. It is available only, if group numbers have been activated in function **Define ROIs**. The resulting list of regions will contain only the regions from the selected group to allow for unambiguous evaluation in function **Evaluate Classes**.
Select	This button opens the **Select Group for Classification** dialog for graphical selection of the group to be classified. It is available only, if the adjacent "Regions in group no." button is activated.
Options	This button opens the **Options for Classification** dialog for setting options for the behavior of the classifier.
Training data	A training data file has to be attached to each function **Classify ROIs**. The **Browse...** button in this area lets you select an existing training data file, whereas the **Edit...** button calls the Training Data Wizard for creating or editing training data files.
Classifier	A classifier has to be attached to each function **Classify ROIs**. The **Browse...** button in this area lets you select an existing classifier, whereas the **Edit...** button calls the Classifier Wizard for creating or editing classifiers.
Assigned training data file	Name of the training data file used to train the currently assigned classifier. Should be identical to the training data file currently attached to the function in order for classifier optimization to work.

2.7.5.9.3 Classify Regions of Interest: Options Dialog

This dialog box is opened by choosing "Options" in the parameter dialog of function Classify ROIs.

It contains the following elements:

Element	Description
Rejection threshold	Sets the minimum classification certainty required for an object to be considered as classified. This value is significant for the following option. Function Evaluate Classes has its own rejection threshold giving you the freedom, for example, to

	automatically store all patterns classified with a certainty below 90% for optimizing the classifier, but to accept results down to a certainty of 80% later (or vice versa).
Automatically save uncertainly classified regions	If this check box is activated, the function will append every region classified with a certainty below the rejection threshold to the training data file attached to this function in its Parameter dialog. The patterns can later be used to optimize the classifier using the **Classifier** command from the Edit menu or the "Edit" button in the "Classifier" area of the parameter dialog.
Maximum size of file	In this edit box the maximum size of the training data file is specified in kBytes. When the file reaches this size, appending of uncertainly classified patterns will stop.

2.7.5.9.4 Classification Features

This section describes the features computed by check function Classify ROIs. The features can be used by check functions like **Sort ROIs**, and **Screen ROIs**.

The following features are computed and added to the features of every region of interest classified by this function:

Feature	Description
Class	Index of the class to which the region is assigned by the classifier.
Class quality	Recognition certainty of the classifier for this region.

2.7.5.10 Evaluate Classes

2.7.5.10.1 Evaluate Classes: Introduction

Function

This function evaluates class information added to regions of interest by classification or comparison functions. It can simply read the information and display or transmit it, but it can also compare the class information of the available regions with a predefined list of required classes, e.g. to compare a line of characters to a prescribed string.

The following functions generate class information:

- Classify ROIs
- Template Matching
- Color Matching

Class information always consists of a class name and a certainty value between 0.0 and 1.0 also computed by the classification or comparison function. The certainty value reflects the fact that classification is inherently a statistical operation. Despite all the assistance given the user by NeuroCheck, classification remains a difficult topic, that requires some background knowledge. A basic introduction to classification and step-wise instructions are given in section "Classification".

Properties

Check function group Analysis

The check function is a decision check function

The check function has target values that can be changed dynamically.

The check function can output result data.

The check function has a target value dialog

Input / output data

Input	Image. List of regions of interest.
Output	List of regions of interest, possibly screened according to class information or certainty. No new data object is created in the data pool.

2.7.5.10.2 Evaluate Classes: Target value dialog

This section describes the settings to be made in the target value dialog of check function **Evaluate Classes**. The dialog is opened by choosing **Target Values** from the **Check Function** menu or the context menu, when check function **Evaluate Classes** is selected.

Element	Description
Verify	If this check box is activated the check function will compare the classes of the regions in its input list to the classes in the target class list below. An object whose class is not contained in the target class list is considered misclassified.
Target classes	List of valid classes or, if **Observe sequence** is activated, required sequence of classes.
Change	Opens the **Edit Class Sequence** dialog for editing the target class list.
Observe sequence	If this check box is activated, the function will compare the classes of the regions in its input list to the target class list in exactly the same sequence, i.e. to verify a character string. If it is not activated, the function will declare objects as misclassified, whose class is not contained in the target class list, regardless of position. In other words: if activated, the functions acts as a string reader, else it only checks for invalid objects.

Rejection threshold	This slider sets the minimum required classification certainty. Objects with a certainty below this value are considered misclassified. Typical values are around 70 or 80%.
Terminate with error	If this option is selected, the function will terminate the current individual check with "not OK" when it finds a misclassified object (according to the target class list and the rejection thresholding).
Delete object	If this option is selected, the function will always return "OK". All misclassified objects will be removed from the list of regions. Thus a classifier can be used as an adaptive object filter.

2.7.5.11 Identify Bar Code

2.7.5.11.1 Identify Bar Code: Introduction

Function

Reads standard bar codes inside a single region of interest. The region of interest should be rectangular and cross the bar code with some room to spare on both ends. The region is scanned parallel to its maximum extension (which may be horizontal or vertical), optionally in the direction of ascending coordinate values (left to right, top to bottom), descending coordinate values (right to left, bottom to top) or both.

Properties

Check function group Analysis

The function is a decision check function

The check function has target values that can be changed dynamically.

The check function can output result data. It sends respectively writes the identified bar code string.

- The function has two dialogs:

  Parameters with parameters for the identification process

  Target values for entering the string to be found on the test piece if you want NeuroCheck to compare the identified code to a prescribed string.

Input / output data

Input	Image. List of regions of interest. Scanning stops after the first region in the list that contained an identifiable bar code of the given type.
Output	None.

2.7.5.11.2 Bar Code Algorithm Parameters

This section describes the settings to be made in the parameter dialog of check function **Identify Bar Code**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when function **Identify Bar Code** is selected.

The dialog has the following elements:

Parameter	Description
Bar code type	Select the type of bar code contained within the region of interest from this list box. Available codes are: • Code 39 • Code 2/5 interleaved • EAN 8 • EAN 13 • UPC version A • UPC version E • PZN • Code 128
Number of characters	Sets the number of characters (mostly digits) contained in the bar code. Only available for Code 39 and Code 2/5 Interleaved.
Detection	Selects the direction of the scanning operation: • **In principal direction:** Scans the region in the direction of ascending coordinate values, i.e. from left to right in horizontal regions, from top to bottom in vertical regions; • **Against principal direction:** Scans the region in the direction of descending coordinate values; • **Both directions:** Scans the region in both directions.
Line distance	Sets the distance of search rays for scanning the region of interest for a readable bar code.
Smoothing	Sets the number of lines on both sides of the search rays to be averaged before reading.
Check sum	If this check box is activated, the function performs a check sum test on the bar code. This is meaningful only if one of the digits of the bar code represents a check sum. Only available for Code 39, Code 2/5 Interleaved and Code 128.

2.7.5.11.3 Bar Code Target Value Parameters

This section describes the settings to be made in the target value dialog of check function **Identify Bar Code**. The dialog is opened by choosing **Target Values** from the **Check Function** menu or the context menu, when check function **Identify Bar Code** is selected.

The dialog has the following elements:

Parameter	Description
Check target code	If this check box is activated, the bar code read is compared to the target string entered below.
Target string	Enter here the bar code string to be present within the region of interest.

2.7.5.12 Identify DataMatrix Code

2.7.5.12.1 Identify DataMatrix Code: Introduction

Function

Reads DataMatrix codes according to the symbology specification 1.01 of the AIM inside a single region of interest. The region of interest should be rectangular and enclose the DataMatrix code with some room to spare on all sides.

Properties

Check function group Analysis

The function is a decision check function

The check function has target values that can be changed dynamically.

The check function can output result data. It sends resp. writes the identified code string.

- The function has several dialogs:

  Parameters for direct access to all parameters of the identification process.

  Wizard for easy first-time configuration.

  Target values for entering the string to be found on the test piece if you want NeuroCheck to compare the identified code to a prescribed string.

Input / output data

Input	Image. List of regions of interest. Scanning stops after the first region in the list that contained an identifiable code.
Output	None.

2.7.5.12.2 DataMatrix Code Algorithm Parameters

This section describes the settings to be made in the parameter dialog of function **Identify DataMatrix Code**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when function **Identify DataMatrix Code** is selected.

The dialog gives direct access to all parameters of the code detection and identification process. Especially for first-time configuration it may be convenient to use the **DataMatrix Wizard** reached by clicking the **Wizard** button.

Parameter	**Description**
Code type	Select from this list the size of the actual data area (including the one module wide frame on each side)
Code color	**Dark:** If this option is selected, NeuroCheck searches for a code with a dark frame. **Light:** If this option is selected, NeuroCheck searches for a code with a light frame.
Code quality	**Good quality**: If this option is selected, code detection is accelerated, because NeuroCheck can skip processing steps for the distinction of code and noise edges. **Poor**: If this option is selected, damaged codes and codes on problematic surfaces can be found more reliably.
Options	This button opens an additional options dialog for settings which are required less frequently.
Code size in pixels	This parameter has to be at least as large as code appears in the image. You can measure the code in an existing image using the rulers and cross-hair of NeuroCheck in manual or live mode.
Reference angle	Select from this list the typical angle of the print in steps of 45°.
Angle range	Set in this box the maximum variation of the code orientation to be expected. The smaller the angle range, the faster the detection.
Undersampling	This parameter sets the step width of the initial search stage. The higher it is, the faster is the detection process, but when set too high the algorithm may miss individual code modules.

Minimum edge height	This parameter sets the contrast required so that NeuroCheck regards the transition between two pixels as an actual edge in the code. If it is set too high, the algorithm may miss edges of the code, if set too low, it may be confused by spurious edges in noisy background.

2.7.5.12.3 DataMatrix Wizard

The **DataMatrix Wizard** provides an easy step-by-step procedure for first-time configuration of DataMatrix code detection and identification. It is reached by clicking on the button **Wizard** in the parameter dialog of check function **Identify DataMatrix Code**.

Each page of the Wizard provides a **New image** button which will cause a new image to be captured from the camera and displayed in the graphics area.

The Wizard leads you through the following configuration steps:

Code color

On this page you have to tell NeuroCheck whether the code is printed dark on light or light on dark by comparing the image of the code to the icons representing the two cases.

Reference angle

On this page you can set the reference angle in which the code should appear and a range around this reference angle. NeuroCheck will look for the code frame at all orientations within this range. The smaller the angle range, the faster the detection process will be and the less susceptible to other edges which may be present in the image. On the other hand, detection may fail if the code deviates from the reference angle more than the range allows.

Edge finding

NeuroCheck has to find the code frame first to determine position and module configuration of the code. On this page, parameters for the edge detection process are configured. The detected edges by applying the current settings are indicated by markers in the image.

NeuroCheck provides two different approaches, a faster one for codes exhibiting high contrast on fairly good surfaces and a slower one which takes into account that there may be spurious edges in the neighborhood of the code which have to be ignored in the detection process.

Also, the minimum edge height can be set, i. e. the gray level difference required by NeuroCheck to accept a brightness change as a code edge.

Undersampling can further accelerate code detection by skipping image pixels, but may result in difficulties for codes with very poor quality.

Code type

On this page you have to tell NeuroCheck the number of square modules along the side of the code. The frames are included in the count, as indicated in the image.

The wizard tries to detect the code type automatically and presents a recommendation in the dialog. Click the button **Apply recommendation** to use this value.

Code size

Here you have to tell NeuroCheck the size of a square guaranteed to enclose the entire code with sufficient quiet zones. Regardless of the general setting in NeuroCheck, this page will always display rulers in its graphics area to help with determining the required size.

The wizard tries to determine the code size automatically and presents a recommendation in the dialog. Click the button Apply recommendation to use this value.

Test decoding

The **New image** button on the final Wizard page not only displays a new image but also executes the entire code detection and decoding sequence. If the code is not decoded properly, you can go back to the previous wizard pages and change the settings.

2.7.5.12.4 DataMatrix Code Target Value Parameters

This section describes the settings to be made in the target value dialog of check function **Identify DataMatrix Code**. The dialog is opened by choosing **Target Values** from the **Check Function** menu or the context menu, when check function **Identify DataMatrix Code** is selected.

Parameter	**Description**
Check target code	If this check box is activated, the identified code is compared to the target string entered below.
Target string	Enter here the code string to be present within the region of interest.

2.7.6 Gauging

2.7.6.1 Introduction to Check Function Group "Gauging"

This section gives some basic information about check function group **Gauging** in NeuroCheck. A more comprehensive introduction to gauging is given in section "Gauging: Introduction".

Geometrical properties

NeuroCheck uses geometrical descriptions of objects for gauging. These can be actually existing properties of the objects, e.g. the coordinates of its center of gravity, but also properties of model geometries, computed from one or several objects.

Every object found by NeuroCheck (e.g. using check function **Create ROIs by thresholding**), owns the following geometrical descriptions:

- **Point**:
  coordinates of the object's center of gravity or the center of a model geometry, if one has been computed for the object;
- **Contour**:
  the actual silhouette of the object (note that an object created by check function **Template Matching** uses the borders of the template as its contour, as there need not be a visually connected silhouette inside the template).

NeuroCheck can also generate the following model geometries, using check function **Compute model geometries** from the Objects category:

- **Line**:
  optimal approximation of an object contour or a line through the centers of a group of objects by a straight line.
- **Circle**:
  optimal approximation of an object contour or the configuration of an object group by a circular arc.

Gauging rules

The applicability of a gauging rule in a certain situation depends on the number of objects to be measured and the geometrical descriptions available for the objects. Measuring the waviness e.g. requires the presence of a contour and a line as model geometry, measuring distances requires two objects etc. Object selection and assignment of gauging rules is done in the Parameter dialog of check function **Gauge ROIs**. The gauging rules available for the various combinations of geometrical descriptions are explained in detail in section "Gauging rules".

2.7.6.2 Gauge ROIs

2.7.6.2.1 Gauge ROIs: Introduction

Function

This check function computes geometrical measurements for one or more regions of interest. It differs from check function **Compute features** in the following respects:

- It cannot compute gray level features, gradients etc.
- It can compute measurements describing relations between several regions of interest, i.e. distances, angles etc. See section "Gauging rules" for a list of all available measurement rules.
- It can compute measurements for model geometries, i.e. for straight or circular approximations of contours. See section "Geometrical descriptions" for more information.
- It is able to work with subpixel precision. See section "Subpixel precision gauging" for more information.

The description of the check function itself is subdivided into the following sections:

- Parameter dialog describes the control elements of the function.
- ROI Selection describes the first page of the Gauging Wizard on which the regions to be gauged are selected.
- Gauging Rule describes the second page of the Gauging Wizard on which the gauging rule to be used on the regions is selected..
- How to define measurements briefly describes the procedure for defining a new measurement.

Properties

Check function group Gauging

The check function has a Parameter dialog.

The function can output result data.

Input / output data

Input	Image. List of regions of interest.
Output	New Measurement list. A new data object is created in the data pool.

2.7.6.2.2 Gauge ROIs: Parameter Dialog

This section describes the parameter dialog of check function **Gauge ROIs**. The dialog is opened by choosing **Parameters** from the **Check Function** menu or the context menu, when the check function **Gauge ROIs** is selected.

The dialog contains the following elements:

Element	Description
Measurement list	Lists all measurements currently defined in this check function with their identification numbers, description text and the gauging rule used to compute the measurement. After selecting a measurement by clicking its identification number it can be edited or deleted using the buttons on the right.
New	Opens the Gauging Wizard for the definition of a new measurement.
Edit	Opens the Gauging Wizard for editing the measurement selected in the measurement list.
Delete	Deletes the measurement selected in the measurement list.
Angle tracing	If this check box is activated, the direction angles of all measurements computed with directed gauging rules will be adjusted according to the last reference angle computed by check function **Determine position**.
Subpixeling	If this check box is activated, all measurements will be computed with subpixel precision, provided the regions involved in the computation contain subpixeling information and the geometrical description selected for the regions allows for a subpixel computation. See section "Subpixel precision gauging" for more information.

2.7.6.2.3 How to Define a new Measurement

This section describes the procedure for generating new measurements using check function **Gauge ROIs**. It is assumed that you inserted the check function into a check and opened its parameter dialog.

Creating a new measurement

1. Choose **New**. The Gauging Wizard appears.
2. Select the regions of interest to be gauged by clicking them with the left mouse button in the image panel. You can select up to three regions of interest, depending on what type of measurement you want to have computed:
    - "Scalar" measurements, like coordinates, radii, areas can only be computed for a single region of interest.
    - Distances between objects and intersections of straight lines require two regions of interest to be selected.

- The angle at a certain point and distances between an object and line intersections require three regions of interest.

3. Select the geometrical description for the selected regions either from the combo boxes on the right or by clicking the regions with the right mouse button. The context menu and the combo boxes offer all geometrical descriptions currently existing for the selected region. **Point** and **Contour** are available for every region, model geometries (like "line", "circle") only if they have been created beforehand for the region. The choice of geometrical description determines the applicability of some measuring rules.
4. Choose **Next**. The **Gauging Rule** page of the Gauging Wizard is displayed.
5. From the **Gauging rule** box choose the measurement to be computed. The choices offered in the box depend on the number of selected regions and the geometrical descriptions available. Please refer to "Gauging rules" for a detailed list of possible gauging rules.
6. If the gauging rule selected involves a direction, the **Angle** edit box will become available. Here you can enter the direction to be used as a counter-clockwise angle from the positive x-axis. For example, if you want a vertical distance to be measured, you would enter "90" in this box.
7. Enter a descriptive text. This is mandatory, you will not be able to leave the Gauging Wizard without entering a text here.

Sections "Gauging a radius" and "Gauging a distance" illustrate this procedure on two examples.

2.7.6.2.4 Gauging a Radius

This section describes how to configure gauging of an object radius in the Gauging Wizard. It is assumed that you inserted check function **Gauge ROIs** into a check and opened its parameter dialog.

Gauging the maximum radius of an object

The maximum radius of an object is defined as the greatest distance from the object's center of gravity to all points on its contour line. It is a scalar measurement, i.e. only one region of interest may be selected for this measuring rule.

1. Choose **New**. The Gauging Wizard appears.
2. Select the region of interest to be gauged by clicking it with the left mouse button in the image panel.
3. Select geometrical description **Contour** for the selected region either from the combo box **Object no. 1** on the right or by clicking the region with the right mouse button and choosing **Contour** from the context menu.

4. Choose **Next**. The Gauging Rule page of the Gauging Wizard is displayed.
5. From the **Gauging rule** box choose **Radius, maximal**. The **Current value** field displays the radius of the selected region.
6. Enter a descriptive text. This is mandatory, you will not be able to leave the Gauging Wizard without entering a text here.
7. Choose **Finish**. The parameter dialog of check function **Gauge ROIs** now displays the newly defined measurement at the bottom of the list.

2.7.6.2.5 Gauging a distance

This section describes how to configure gauging of the maximal distance between two objects in the Gauging Wizard. It is assumed that you inserted check function **Gauge ROIs** into a check and opened its parameter dialog.

Gauging the maximum distance of two objects

The maximum distance of two objects is defined as the length of the line between those points on the contours of the two objects farthest apart.

1. Choose **New**. The Gauging Wizard appears.
2. Select the two regions of interest to be gauged by clicking it with the left mouse button in the image panel.
3. Select geometrical description **Contour** for both regions either from the combo boxes **Object no. 1** and **Object no. 2** on the right or by clicking each region with the right mouse button and choosing **Contour** from the context menu.
4. Choose **Next**. The **Gauging Rule** page of the Gauging Wizard is displayed.
5. From the **Gauging rule** box choose **Distance, maximal**. The **Current value** field displays the distance of the selected regions.
6. Enter a descriptive text. This is mandatory, you will not be able to leave the Gauging Wizard without entering a text here.
7. Choose **Finish**. The parameter dialog of check function **Gauge ROIs** now displays the newly defined measurement at the bottom of the list.

2.7.6.3 Derive Measurements

2.7.6.3.1 Derive Measurements: Introduction

Function

This function performs calculations on measurements computed by check function **Gauge ROIs**. Details on the possible calculations and the editing procedure are given in the following sections.

Properties

Check function group Gauging

The check function has parameters that can be changed dynamically.

The function can output result data.

The check function has a parameter dialog with the following elements:

Parameters

Element	Description
Derived measurements	This list displays all derived measures defined so far.
New	This button opens the **Edit Measurement Computation** dialog, where you can enter the parameters for a new calculation.
Edit	This button opens the **Edit Measurement Computation** dialog for altering the parameters of the currently selected computation.
Delete	This button deletes the measurement currently selected in the **Derived measurements** field.

Input / **output data**

Input	Measurement list.
Output	Measurement list with the values computed by this check function. A new data object is created in the data pool.

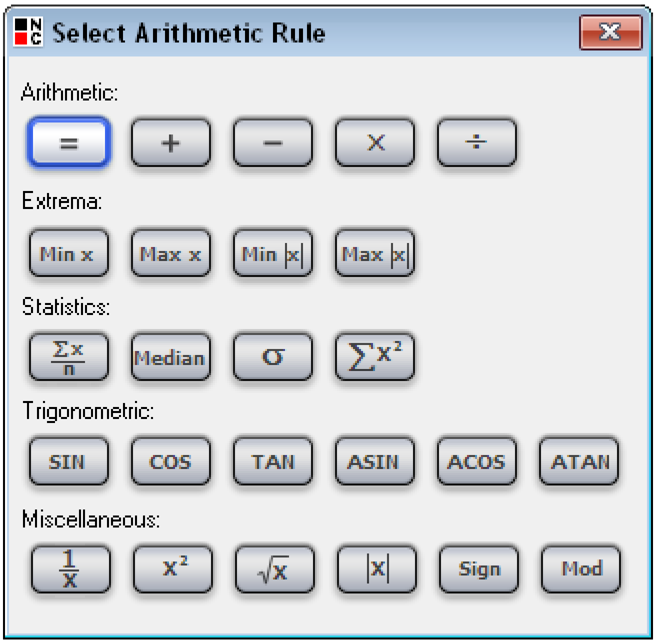
You can copy a value without change from the input to the output list by selecting the measurement and defining the rule **Assignment**.

2.7.6.3.2 Arithmetic Rules for Measurement Calculations

This section lists the arithmetic rules available for computations with measurements in check function **Derive measurements**.

Arithmetic Rules dialog

The dialog is designed like a calculator:



The availability of a calculation rule depends on the number of selected input values. The optional fixed value counts like a measurement from the input measurement list. You can choose one of the following rules:

Arithmetic rules for one measurement

If one measurement (or the fixed value) has been selected for calculations, the following calculations rules are available:

- Assignment: Copies the input measurement without change and without doing any calculations.
- Sine: The result is the sine of the selected measurement.
- Cosine: The result is the cosine of the selected measurement.
- Tangent: The result is the tangent of the selected measurement.
- Arcsine: The result is the arcsine of the selected measurement.
- Arccosine: The result is the arccosine of the selected measurement.
- Arctangent: The result is the arctangent of the selected measurement.

- Inverse: The result is the inverse value of the selected measurement.
- Square: The result is the square of the selected measurement.
- Root: The result is the square root of the selected measurement.
- Absolute value: The result is the absolute value of the selected measurement.
- Signum: The result is 1 if the input measurement is a positive number; the result is -1 if the input measurement is a negative number; and 0 if the input measurement is 0.

Arithmetic rules for more than two measurements

If two or more measurements (including a fixed value) have been selected, the following arithmetic rules are available:

- Addition: the result is the sum of all selected values.
- Multiplication: the result is the product of the selected values (Be careful when using more than two measurements, possible data overflow).
- Minimum value: the result is the smallest of all selected values.
- Maximum value: the result is the largest of all selected values.
- Minimum abs. value: the result is the minimum of the absolute values of all selected measurements.
- Maximum abs. value: the result is the maximum of the absolute values of all selected measurements.
- Average: the result is the arithmetic mean of all selected values.
- Median: the result is the median of all selected values, i.e. the value, below which there are as many values than above.
- Standard deviation: the result is the empirical standard deviation of all selected values.
- Square sum: the result is the sum of the squares of all selected values.
- Square root: the result is the square root of the sum of all selected values (this allows computation of the Pythagorean proposition by using square sum in the first instance of this function, then square root on a single value in a second instance).

Additional arithmetic rules for exactly two measurements

If exactly two values (including a fixed value) have been selected, the following rules are available in addition to the ones above:

- Subtraction: the result is the difference between the selected values, whereby the last value from the list is subtracted from the first.
- Division: the result is the first value from the list divided by the second.
- Modulo: Finds the remainder of a division of two integers. For example, this function can be used to limit the angle value of the selected measurement (incl. a fixed value) to a value range between 0 and 359.

2.7.6.3.3 Specifying a Measurement Calculation

This section describes the procedure for define a measurement calculation using check function **Derive measurements**. It is assumed that you have appended the function to a check and opened its parameter dialog.

Choose **New** to define a new calculation or **Edit** to edit the calculation of the value currently selected in the list of derived measurements. In both cases, the **Edit Measurement Calculation** dialog will be displayed. If you choose **New**, no values will be preselected, else it will show the current configuration of the derived measurement selected for editing.

Now proceed as follows:

1. Select the measurements to be used in the calculation by clicking the boxes in front of the identification numbers in the **Input measurements** list. You can instruct NeuroCheck to include an additional fixed value in the calculation by checking the **Fixed value** box and entering a value.
2. Select the computation to be performed by pressing the **Arithmetic rule: Change** button. The available rules depend on the number of selected values (including a possible fixed value). See section "Arithmetic rules for measurement calculations" for a list of available calculation rules.
3. Select **Exchange arguments** to invert the order of the selected input measurement values. Only available on **Division** and **Subtraction**!
4. Select whether you want the absolute value of the result to be computed.
5. Enter a descriptive text for the computed value (this is not mandatory, but recommended to be able to identify the values other than by their identification numbers).
6. Choose **OK**. The dialog is closed. If you opened it by choosing **New** in the function parameter dialog, the list of derived measurements will have the newly defined measurement appended.

2.7.6.4 Combine Measurement Lists

2.7.6.4.1 Combine Measurement Lists: Introduction

Function

Concatenates measurement lists generated by different functions. The measurement list generated first in the data pool is appended to the one generated second.

Properties

 Check function group Gauging

 The function has a parameter dialog

The parameter dialog contains the following element:

Element	Description
Offset of indices	Value by which the identification numbers in the first input measurement list will be increased. The value has to be at least as high as the highest identification number in the second measurement list. Higher offsets enable you to add measurements to the second list later. Only available on disabled checkbox "Use automatic offset"!
Use automatic offset	Determines the offset automatically to prevent potentially errors in combination of indices.

Input / output data

Input	Measurement list 1. Measurement list 2.
Output	Measurement list 2 will be supplemented by the measurements of list 1. No new data object is created in the data pool.

The check function does not generate a new measurement list, but alters the second input list instead. Therefore, no new data object appears in the data pool. The sequence of the input lists corresponds to the sequence in which they have been created in the data pool. Automatic input configuration will therefore use the result list. If the unchanged input list is to be used in subsequent functions, you will have to select this list in the Input configuration dialog box.

2.7.6.5 Calibrate Measurements

2.7.6.5.1 Calibrate Measurements: Introduction

Function

This check function determines a conversion factor to compute metrical dimensions from image pixels and converts all measurements in its input measurement list according to this factor.

The check function differs from check function **Calibrate pixels** in the following respects:

- The function requires a measurement list as input.
- Conversion factors can be entered manually as well as determined automatically from the relation of any measurement computed by **Gauge ROIs** or **Derive Measurements** to a predefined reference value.
- Conversion factors can be determined as averages of several different measurements, e.g. the height and the width of a part.

Properties

Check function group Gauging

The check function has a Parameter dialog.

The check function can output result data (the converted measurement list)

Input / output data

Input	Measurement list.
Output	The measurements in the input list are altered. No new data object is created in the data pool.

2.7.6.5.2 Calibration Modes

This section explains the different calibration modes of check function Calibrate measurements.

The check function can be used in one of the following modes:

Offline

In this mode, a new conversion factor is computed only upon leaving the parameter dialog with **OK**. This value will then be used for all subsequent measurements. All checked measurements (green checkmark) will use their own factor. All unchecked measurements are using an average factor, calculated of all checked measurements. This mode is useful under the following conditions:

- The calibration refers to an object of stable size contained in the image scene.
- The capturing conditions are stable, making new calibrations unnecessary except under certain well-defined circumstances and can therefore be performed interactively, e.g. after a change in production.

Online

In this mode, the conversion factor is computed every time the check function is executed, i.e. every time the check runs. An average of all calibration factors is applied to all measurements. This mode is useful under the following conditions:

- The calibration refers to an object of stable size contained in the image scene.
- The capturing conditions are not stable; for example, the distance between camera and object may very slightly. In this case a new calibration is necessary after every image capturing operation, i.e. each time the check runs.

Manual

In this mode, a conversion factor can be entered manually, just like in check function **Calibrate pixels**, the only difference being that the check function uses the factor to convert the measurements in its input list immediately. This fix value will be applied to all measurements.

2.7.6.5.3 Using Check Function "Calibrate Measurements"

This section introduces different ways of employing check function **Calibrate measurements** depending on the calibration mode. The check function needs a measurement list as input. Therefore it requires check function **Gauge ROIs** to be executed beforehand; this in turn requires definition of regions of interest and an object search.

The check function should therefore be used in one of the following ways:

- The check function can be put into a separate check used only to determine the conversion factor; this check will be deactivated in automatic mode using the Check properties dialog box. Every other check of the check routine will then have to contain check function Calibrate pixels in order to introduce the conversion factor determined by check function **Calibrate Measurements** into the other checks. This method will usually be applied, if the conversion factor is stable for long periods of time (e.g. several days) and needs to be checked only rarely.
- The check function can also be put into every check which performs gauging operations. This method is useful, if the conversion factor changes more often, perhaps for every single test piece, e.g. because of varying distances between object and camera caused by positioning inaccuracies. The advantage of this method lies in its accuracy: the conversion factor is determined for every single image captured from the camera. On the other hand, every such calibration requires searching for the reference objects and determining their measurements.
- If there are several measurements to be determined and only one is used for calibration, it is not necessary, to perform a separate object search for the measurement used for calibration and the other measurements. Instead, check function **Calibrate measurements** can convert the measurement list directly.

2.7.6.6 Check Allowances

2.7.6.6.1 Check Allowances: Introduction

Function

This function compares measurements created by **Gauge ROIs** or **Derive measurements** with preset values. As soon as the value of one or more measurements is outside the allowed range, the function creates a target value failure (not OK).

A measurement is considered correct if:

```
(nominal value + lower allowance) <= measured value <= (nominal value + upper
allowance)
```

The result will be presented in an image which contains all measurements. The measurement highlight color differs depending on the result. The default color settings are as follows:

- Measurements within range of tolerance = green
- Measurements outside range of tolerance = red
- Not checked (deactivated) measurements = blue

You can personalize all colors in dialog **Software Settings**. Open Dialog via main menu **System > Software Settings**.

Properties

Check function group Gauging

The function is a decision check function.

The check function has target values that can be changed dynamically.

The check function can output result data.

The check function has a target value dialog.

The target value dialog contains the following elements:

Element	Description
Verify target values	**Enabled**: The target values of this check function will be used for the result evaluation (O.K. / N.O.K.). **Disabled**:

	If this check box is deactivated, the execution of this check functions will not stop and return "N.O.K." in case of a target value failure, but each measurement will be marked as "N.O.K.". The check function has no decision function and it is not part of result evaluation.
Allowance specification	This field displays the available measurements. A measurement is compared to a nominal value if the box in front of the identification number is checked. Clicking the box with the left mouse button toggles the state of the box. Clicking the identification number with the left mouse button selects the measurement for editing target values. If you want to change the values in the table, please edit the values in group box **Allowances for measurement no. …**.
Copy	Copies values of the selected row of the specification table to clipboard.
Paste	Inserts values from clipboard in the selected table row. Present values will be overwritten.
Teach all	The current values (from input measurement list) of all measurements are applied as nominal values for each measurement. Present values will be overwritten.
Method	**Absolute:** If active, the allowances are added to the nominal value as entered. **Percentage:** If active, the entries in the fields **Upper allowance** and **Lower allowance** are treated as percentages of the nominal value. The values actually added to the nominal value are computed by multiplying these percentages with the nominal value (which, therefore, should not be 0).
Nominal value	Here you can enter the specified nominal value for the respective measurement.
Upper allowance	Here you can enter the upper allowance to specify how far the real value may exceed the nominal value.
Lower allowance	Here you can enter the lower allowance to specify, how far the real value may fall short of the nominal value. Please don't forget to type the negative sign.

If group box **Allowances for measurement no. …** contains unusual values, you will get a hint in red letters right hand to the input field. This may be in case, if you use for both allowances (upper and lower allowance) the same negative or positive prefix.

Input / output data

Input	Image, for display only. Measurement list.
Output	None.

2.7.7 Position Adjustment

2.7.7.1 Introduction to Check Function Group "Position Adjustment"

Check function group **Position Adjustment** comprises functions for automatic adjustment of the positions of regions of interest according to the position of reference objects.

Position parameters

To fully compensate for position shifts, the following parameters are necessary:

- Translation in horizontal direction (X offset).
- Translation in vertical direction (Y offset).
- Rotation angle (this parameter can also be used by check function **Gauge ROIs** for compensating rotation in direction dependent measurements).
- Pivot point for the rotation.

These parameters are computed by check function **Determine position of ROIs**. The computation can be distributed at will over different objects. All parameters can be computed from the same object or each from a different object. Alternatively, the parameters can be calculated from the properties of a whole group of objects. Each parameter may only be computed once within the same instance of the function.

Scope of the position parameters

Because of the very clearly defined application of these parameters, they are stored in a data structure global to the individual check in which the function is located. This has the following consequences:

- It is possible to compute the parameters sequentially in several instances of check function **Determine position of ROIs**. It may, for example, be necessary to first compensate for a horizontal offset to be able to find the reference object for the vertical offset at all.
- When computing parameters sequentially it may be necessary to reset parameters explicitly. Without a reset for the x offset parameter in the above example, the x offset would be applied twice.

To re-use the position parameters in a different individual check you can use check function **Set Position Parameters**.

2.7.7.2 Determine position of ROIs

2.7.7.2.1 Determine Position of ROIs: Introduction

Function

This function computes translation and rotation of the regions of interest in its input list, relative to a reference position. The computed position parameters can be used by check function **Position ROIs** to adjust the position of other regions of interest. See section "Introduction to position adjustment" for an explanation of the positioning parameters. Section "Structure of a position invariant check" describes a typical configuration for the use of position adjustment in a check.

Properties

Check function group Position Adjustment

The function generates feature information. See section "Positioning features" for a list of features computed by this function.

The function can output result data. It sends or writes the determined position deviations.

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Image panel	Displays the regions selected for computing the position parameters.
Reference	Lists the parameters computed by this instance of the function and their current values. Which parameters are computed is determined during teach-in on page **Select Transformation(s)** of the Positioning Wizard.
Teach...	Opens the Positioning Wizard for computing a new reference position.
New image	Triggers background execution of the check so that position parameter teaching can refer to the object currently in front of the camera.

Input / output data

Input	Image. List of regions of interest.
Output	The check function changes the position parameter set global to the individual check the check function is located in. Check functions **Position ROIs** and **Gauge ROIs** always use the most recently computed set of position parameters. No new data object is created in the data pool.

2.7.7.2.2 Structure of a Position Invariant Check

This section describes a typical configuration for using the position adjustment functions in a check.

The presence of regions of interest is a prerequisite for using position adjustment. Using manually defined regions of interest, though possible in principle, would not be meaningful, because such regions never change their position relative to the image. To configure the functions, an image containing the check piece at the reference position is therefore required.

The basic structure is always as follows:

1. Searching an object, which can be found reliably and is suitable for determining changes in the position of important objects in the image. If the image may be rotated, the object should allow calculation of the rotation angle. Oblong, asymmetrical objects are best suited for this purpose, square or circular objects least well (although the polar distance algorithm will do an excellent job even then; but it is much more computation-intensive than using the principal axis).
2. Determination of position parameters according to position and orientation of reference objects. This is described in detail in section "Configuring the position parameters".
3. Definition of additional regions of interest for searching objects, whose position depends on the reference objects.
4. Positioning these regions using the position parameters.

2.7.7.2.3 Configuring the Position Parameters

This section describes step by step how to set the reference position parameters. In the parameter dialog of check function **Determine position of ROIs** the following steps are necessary to determine the position parameters:

1. Choose **Teach** to open the Positioning Wizard. Activate all the check boxes on the **Select Transformation(s)** page and ascertain that below each check box the **Calculate** button is selected.
2. Choose **Next**. The **X Offset** page is displayed. Select the region or group of regions to be used as reference for the x position and the reference point. Note that **Left edge** and **Right edge** refer to the leftmost or rightmost points of a single region, but to the center of gravity of the leftmost or rightmost object of a group.
3. Choose **Next**. The **Y Offset** page is displayed. Select the region or group of regions to be used as reference for the y position and the reference point. Most of the times this will be the same object as for the x position, but it can be a completely different one. Note that **Top edge** and **Bottom edge** refer to the top or bottom points of a single region, but to the center of gravity of the top or bottom object of a group.

4. Choose **Next**. The **Rotation angle** page is displayed. Select the region or group of regions to be used for computing the rotation angle.

   Some hints:

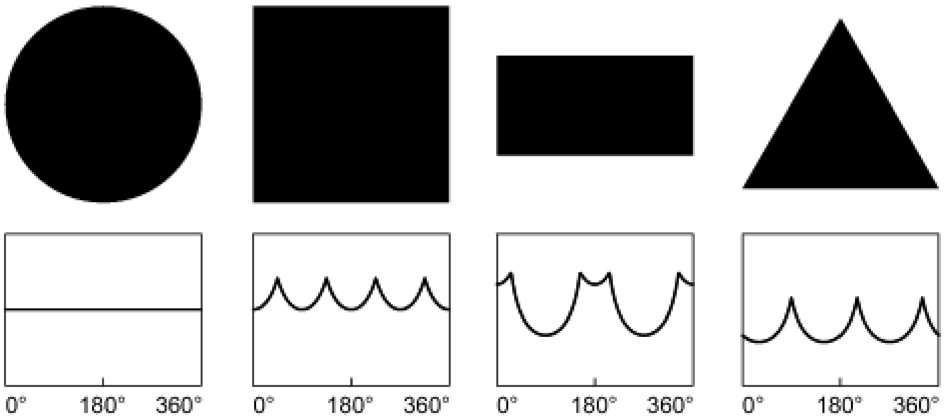
   - Long, asymmetrical objects are best suited as reference objects for rotation.
   - For completely or almost symmetrical objects, the computation of a full circle orientation ("-180° to +180°") is very difficult if not impossible.
   - For near square objects use the **Polar distance** algorithm, which is slower than the **Principal axis** method but much more precise on this kind of objects (because the principal axis may move considerably due to small defects on the contour of such objects).

5. Choose **Next**. The **Pivot** page is displayed. Select the region or group of regions to be used as the pivot. The function always uses the center of gravity of the selected group or region.

During execution of the check routine the check function calculates position and orientation of the corresponding objects in the current image and compares them to the stored values. The deviations are used by function **Position ROIs** to adjust the position and orientation of manually defined regions of interest. Of course, the objects selected as references need to be found reliably for these functions to work correctly. If the image scene changes its overall structure completely, no meaningful comparison of positions and parameters will be possible, causing the function to stop with an error.

Example for polar distance projection

The following images are taken from the reference book **Industrial Image Processing** (1999, ISBN 3-540-66410-6), published by Springer Verlag (Berlin Heidelberg New York). They show polar distances for some simple geometric objects.



Circle Square Rectangle Triangle

2.7.7.2.4 Positioning Features

This section describes the features computed by check function **Determine Position**. The features can be used by check functions like **Sort ROIs**, **Screen ROIs**, and **Classify ROIs**. The following features are computed and added to the features of every region of interest in the input list of this function (even regions not used for the position computation):

Feature	Description
X offset	Offset in x direction of the current image to the stored reference point.
Y offset	Offset in y direction of the current image to the stored reference point.
Pivot X	X coordinate of current pivot point.
Pivot Y	Y coordinate of current pivot point.
Rotation angle	Rotation angle of current image relative to reference image.

2.7.7.3 Position ROIs

2.7.7.3.1 Position Regions of Interest: Introduction

Function

This function adjusts position and orientation of regions of interest according to position parameters calculated by check function **Determine position of ROIs** or by check function **Set position parameters** to compensate for variations in position and orientation of image objects.

The function always uses the set of position parameters most recently computed within the check. Therefore, the position parameters are not explicitly configurable as input data. Section "Structure of a position invariant check" describes a typical configuration for using position adjustment in a check.

The effect of the possible combinations of these parameters is in section "Effect of the Rotation Parameters".

Properties

Check function group Position Adjustment

The function has a parameter dialog

The parameter dialog consists of the following elements:

Element	Description
Object's center of gravity	If this box is checked, each region of interest in the function's input list is rotated around its own center of gravity, according to the computed rotation angle.
Prescribed pivot	If this box is checked, each region of interest in the function's input list is moved

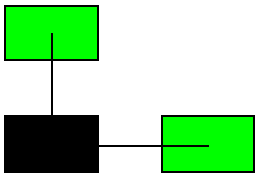
on a circle around the pivot defined in check function **Determine position of ROIs**.

Input / output data

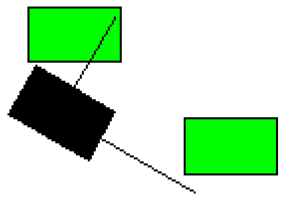
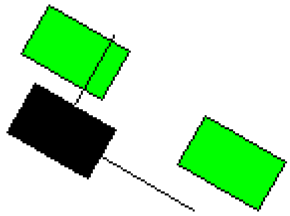
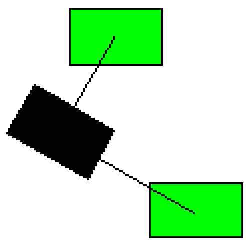
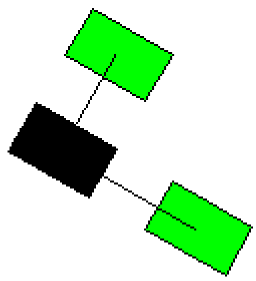
Input	Image. List of regions of interest.
Output	List of regions of interest, shifted and rotated according to the position parameters. A new data object is created in the data pool.

2.7.7.3.2 Effect of the Rotation Parameters

The black object is used as reference object for the rotation angle in the following images. Its center of gravity is also used as the pivot. The green rectangles represent manually defined regions of interest, whose position is adjusted according to the orientation of the reference object.



Rotating the reference object clockwise by 30° has the following effect, depending on the settings of the rotation parameters:

Prescribed pivot	Center of gravity	Effect	Image
off	off	Regions are moved solely according to an eventual offset in x and y direction	
off	on	Regions are rotated in place, then possibly moved in x and y direction	
on	off	Regions are moved on a circle around the reference object's center of gravity, but keep their own orientation	
on	on	Regions are moved in a circle around the reference object's center of gravity. In addition their own orientation is changed according to the rotation angle.	

2.7.7.4 Set Position Parameters

2.7.7.4.1 Set Position Parameters: Introduction

Function

Use this check function to set data for positioning. This can be used for the following applications:

1. Example:
   You want to transfer the position parameters derived from one check (by check function **Determine Position**) to another check without recalculation. For this, have the check function **Determine Position** write its results into five Register cells and, in the other individual check, have the check function **Set Position Parameters** clear its parameters for these register cells.
2. Example:
   You want to move or rotate an ROI by a known, dynamic value. The value was determined in the previous check function, e.g. **Gauge ROIs** . You can accomplish this by writing the results to vacant registers cells and clearing the desired parameters from **Set Position Parameters** for these register cells.
3. Example:
   You are using only fixed values that you want to set in the parameter dialog.

For example 1 and 2 you can also manipulate values that have previously been calculated by other check functions with other register check functions in the registers.

Properties

Check function group Position Adjustment

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog

Input / output data

Input	None
Output	None

2.7.8 Tools

2.7.8.1 Introduction to Check Function Group "Tools"

The check functions in group **Tools** fulfill general utility tasks. Most of them can be used without an image being present in the data pool. In this category you will also find the check functions for manipulation the Data tray.

2.7.8.2 Delay Execution

2.7.8.2.1 Delay Execution: Introduction

Function

Execution of the check routine is delayed by an adjustable time interval. This can be necessary if the check routine has to wait for a certain time until the work piece to be inspected has come to rest. Another example for the application of this check function is the construction of simple communication protocols together with the functions **Read Process Input Bit** and **Set Process Output Bit**.

Properties

 Check function group Tools

 The check function has parameters that can be changed dynamically.

 The check function has a parameter dialog.

The parameter dialog contains the following parameters:

Element	Description
Delay time	Delay in milliseconds, max. 9999

Input / output data

Input	None
Output	None

2.7.8.3 Comment

2.7.8.3.1 Comment: Introduction

Function

This (pseudo) check function contains no functionality. It is used to structure and document extensive check routines or checks. By entering a user-defined check function name, any comment can be displayed directly in the check routine structure.

For this, left-click the check function name twice (no double-click) and substitute your own comment for the default text.

> Also add an eye-catching background or font color for this check function. For these settings use the **Check Function Properties ▸Manual Mode ▸Explorer Display** dialog.

Properties

Check function group Tools

- This check function has no parameters.

Input / output data

Input	None
Output	None

2.7.8.4 Control Device

2.7.8.4.1 Control Device: Introduction

Function

This check function is capable of changing the property of a device or calling a device function. You can use this for instance to specifically change a camera parameter like the exposure time within a running check routine.

> The availability of the properties and functions depends on the driver implementation. For example, the so called "Image stack acquisition" is only available for the driver of the NeuroCheck NCF and NCG digital camera series.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Device	Select here the device that you want to control. The device has to be configured in the NeuroCheck device manager.
Mode	**Set property**: Select this mode to change the value of a device property. **Execute method**: Select this mode to call a device function.
Property/method	Select here the property resp. the function.
Value	Enter here the new value to be set (only available in mode **Set property**).

Input / output data

Input	None
Output	None

2.7.8.5 Calibrate Pixels

2.7.8.5.1 Calibrate Pixel Size: Introduction

Function

This function allows you to enter a conversion factor to compute metrical dimensions from image pixels. A physical unit for the conversion factor has been deliberately left out, because pixels might correspond to kilometers (in satellite images e.g.), inches or millimeters (for technical objects) or even micrometers (for microscopic images).

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog with the following parameters:

Element	Description
Pixel dimension	Here you can enter how many of the preferred physical units (e.g. millimeters or inches) correspond to one image pixel.

Input / output data

Input	None.
Output	None. Every image analysis function uses the conversion factor most recently entered. No new data object is created in the data pool.

2.7.8.6 Execute Process

2.7.8.6.1 Execute Process: Introduction

Function

This check function starts an external process or executes a console command. You can use it to start external programs or batch files, e.g. for management or backup jobs.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Process type	**External program**: In this mode, the check function calls an external program or file. The object called can be selected by using the **Browse...** button. **Console command (shell)**: In this mode, the check function issues a console command to the operating system. The command must be entered in the text box.
Command line arguments	Here you can enter optional command line parameters that will be passed to the external program upon being called. Please check the documentation of the external program for available parameters. Only available for process type **External program**.

Process call behavior	**Synchronous**: In this mode, the check function waits for the called process to be completed. Only in this mode you can evaluate error messages of the called process that may have occurred. **Asynchronous**: In this mode, the check function does not wait for the called process to be completed. The execution of the check function is continued immediately.
Time out [ms]	Here you can set the maximum waiting time that the check function will wait for the completion of the called process. After this time, the check function will terminate with an error.
Ignore exit code	**Selected**: The return value of the process called will be ignored. **Deselected**: The return value of the process called will be evaluated. If the process returns an error (exit code not equal to 0), the check function will terminate with an error.
Skip execution in manual mode	When this option is active, the check function will not be executed in manual mode but only in automatic mode.
Windows state	Here you can set the status of the window where the console command will be executed. Only available for process type **Console command**.

Input / output data

Input	None
Output	None

2.7.8.7 Modify Data in Register

2.7.8.7.1 Modify Data in Register: Introduction

Function

This check function changes contents of data registers. You can define changing rules in a list that will be executed in the order listed.

Possible applications

This very powerful check functions can be used in various ways: Examples:

- Initializing and incrementing execution counters
- Copying data among input, output and empty registers

- Combination of the contents of several registers
- Calculation of sums, averages or statistical values for characteristics or measurements etc.
- Conversion of various data types within the registers
- Logical operations among check results (AND, OR)

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

Input / output data

Input	None
Output	None

2.7.8.8 Verify Data in Register

2.7.8.8.1 Verify Data in Register: Introduction

Function

This check function checks the contents of one or more register cells against target values. This way a decision between "OK" or "not OK" can be made using the contents of register cells (decision function). As soon as at least one of the checks does not comply with the target values, the check function is aborted with not OK. You can define the checks for several register cells using a list.

Possible applications

- Checking values that were used for calculations with check function **Modify data in register** for adherence to target values.
- Checking check function results for adherence to target values that don't have an explicit decision function.
- Checking dynamic input data for compliance with value ranges or plausibility.

Properties

Check function group Tools

The check function is a decision check function.

The check function has target values that can be changed dynamically.

The check function has a target value dialog.

Input-/Output data

Input	None
Output	None

2.7.8.9 Start Timer

2.7.8.9.1 Start Timer: Introduction

Function

The two check functions **Start Timer** and **Stop Timer** enable you to measure and check the interval between two instants of time during check routine execution. Thus you can limit the maximum execution time of the check routine or individual checks, if necessary.

This check function determines the current system time accurate to the millisecond and saves it to a time buffer.

Properties

Check function group Tools

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Timer ID	Here you can set the timer ID. You can manage up to 100 different time buffers independently.

Input-/Output data

Input	None
Output	None

2.7.8.10 Stop Timer

2.7.8.10.1 Stop Timer: Introduction

Function

The two check functions **Start Timer** and **Stop Timer** enable you to measure and check the interval between two instants of time during check routine execution. This allows you to limit the maximum execution time of the check routine or one of the checks, if necessary.

This check functions stops the timer and calculates the time that has passed since check function **Start Timer** has been started, accurate to the millisecond.

It is possible to compare this time to a definable maximum allowed time. If this target time was exceeded in automatic mode, the check function will yield not OK (decision check function!).

Upon execution of this check function, the starting time will be preserved in the time buffer. Thus it is possible to monitor the same timer with several instances of this check function at various points in the check routine.

Properties

Check function group Tools

The check function is a decision check function.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Timer ID	Here you can set the timer ID that you have set in the check function instance **Start Timer** which you want to compare to.
Verify	Activates the verification of complying with the maximum time. If this switch is deactivated, the check function registers the current time but does not do any checking.
Maximum time	Here you can enter the maximum time in milliseconds that is allowed to have passed since the check function instance **Start Timer** has been executed. If this time is exceeded in automatic mode, the check function will yield not OK.
Restart timer	If this switch is activated, the time buffer will be overwritten with the current system time after executing this check function.

Input-/Output data

Input	None
Output	None

2.7.8.11 Data Tray

2.7.8.11.1 Data tray: Introduction

Data tray concept:

The NeuroCheck data tray is a global data storage for run-time data objects for using them in a different check or a different check routine.

All data objects of the data pool can be managed in the data tray:

- Type Images
- Type Lists of ROIs
- Type Measurement lists
- Type Histograms

Functionality:

The data tray can contain up to 100 data objects of each of the above-mentioned types. The content of each storage position can be directly accessed via an index no. (0 - 99). Data tray access is only possible using check functions. These check functions can be found (save two exceptions) in the "Tools" function group, sub-category "Data Tray". The data tray can be accessed as follows:

- Copy a data object from the data pool of a check into the data tray.
  The pertaining check functions copy the data object from the data pool of the check and insert it into in the global data tray using the given index number.
- Copy a data object from data tray into the data pool of an individual check.
  The pertaining check functions copy the data object with the selected index number from the global data tray into the check's data pool where it is then available as a new data object. The check function aborts with an error message if there is no valid data object filed under the selected index number.
- Moving or deleting data objects within the data tray.
  Data objects can be moved to a different index no. within the data tray (shift register functionality) or cleared from the index (to make sure they can only be read out once) using the check function **Control Tray**.
- Direct access to image data tray.
  The check functions **Transfer Image to Tray** and **Transfer Image** from the check function category Image Acquisition also have access to the data tray but can only access data objects of the "Images" type.
  The check function **Transfer Image to Tray** transfers captured camera images directly to the data tray.
  The check function **Transfer Image** (with source: **Image Tray**) copies an image from the data tray to the data pool for further processing.
- Visualizing the data tray: In the **Tools** menu, choose **Data Tray** to display a **Data Tray Viewer** (display dialog), allowing you to view all currently available data objects.

Overview of all check functions with access to the Data Tray:

Copy objects to Data tray:

- Copy Image to Tray
- Copy ROIs to Tray
- Copy Measurement List to Tray
- Copy Histogram to Tray

Copy objects from Data tray:

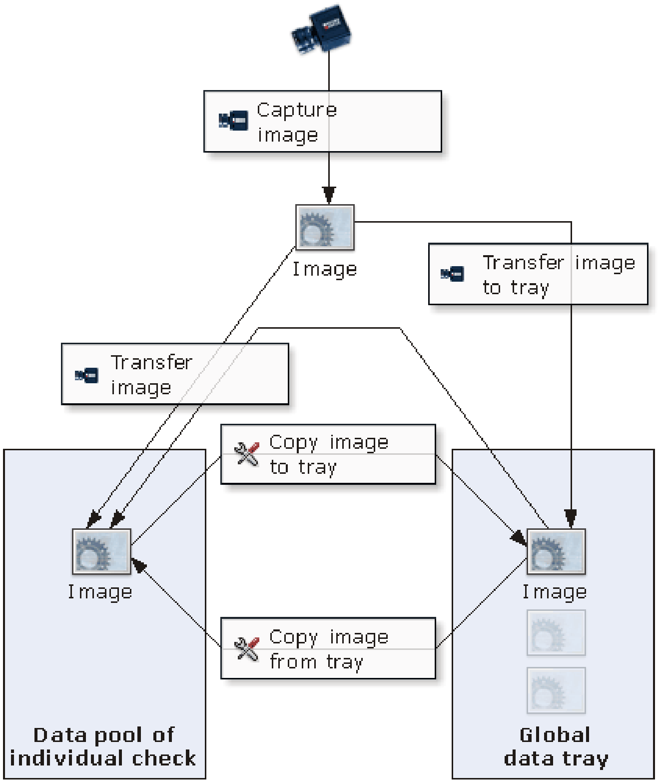
- Copy Image from Tray
- Copy ROIs from Tray
- Copy Measurement List from Tray
- Copy Histogram from Tray

Move or delete (reset) saved objects in Data tray:

- Control Tray

Direct Access to Image tray:

- Transfer Image to Tray
- Transfer Image



How to use the data tray in NeuroCheck:

Typical areas of use for the data tray are the following:

- Quick image capturing for data objects of the "Image" type:
  The check function **Transfer Image to Tray** transfers the most recently captured image to the image tray, from where it can be transferred to NeuroCheck memory for further processing using the check function **Transfer Image**. This check function is typically used in the start actions of a check routine to quickly record a number of images to be processed while the test piece is already moved out of the inspection station.
- Transferring data objects among various checks.
- Transferring data objects between different cycles of a check routine.
- Transferring data objects between different check routines after check type selection.

> ⚠ The data tray objects are only stored in RAM. At the end of the program, all data tray data is lost.

2.7.8.11.2 Copy Image to Tray

2.7.8.11.2.1 Copy Image to Tray: Introduction

Function

This check function copies the input image from the data pool of the single check to the specified index of the global data-tray.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Destination index	Select here the zero-based index in the global tray where the image will be stored. Index range: 0-99

Input-/Output data

Input	Image.
Output	None.

2.7.8.11.3 Copy Image from Tray

2.7.8.11.3.1 Copy Image from Tray: Introduction

Function

The check function copies the input image with the specified index number from the global data-tray to the data pool of a single check. The image will then be available in the data pool as a new data object from type "Image". The function will abort with an error, if the specified index does not contain an image.

This check function is very similar to the functionality of the check function **Transfer image**, if the data source **Image Tray** is selected.

The differences are:

- The setting **Simulate Image Capture** has no effect to the check function **Copy Image from Tray**.
- Using the check function **Copy Image from Tray** you receive the full image always. In the check function **Transfer image** you can select an image section.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Copy from index No.	Select here the zero-based index in the global data-tray from where the image will be copied. Index range: 0-99
Reset after copy operation	**Checkbox enabled:** If this checkbox is activated, then the object at the specified index will be reset/deleted after the image was copied to the data pool. This avoids using the same image a second time inadvertently. **Checkbox disabled:** After the copy has finished, the data object remains in the data tray at the specified index, so that you can reuse it.

Input-/Output data

Input	None.
Output	Image. A new data object is created in the data pool.

2.7.8.11.4 Copy ROIs to Tray

2.7.8.11.4.1 Copy ROIs to Tray: Introduction

Function

This check function copies the list of ROIs from the data pool of the single check to the specified index of the global data-tray.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Destination index	Select here the zero-based index in the global tray where the ROIs will be stored. Index range: 0-99

Input-/Output data

Input	List of ROIs.
Output	None.

2.7.8.11.5 Copy ROIs from Tray

2.7.8.11.5.1 Copy ROIs from Tray: Introduction

Function

The check function copies ROIs with the specified index number from the global data tray to the data pool of a single check. The ROIs will then be available in the data pool as a new data object from type "List of ROIs". The function will abort with an error, if the specified index does not contain a list of ROIs.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

This check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Copy from index No.	Select here the zero-based index in the global data-tray from where the ROIs will be copied. Index range: 0-99

Reset after copy operation	**Checkbox enabled:** If this checkbox is activated, then the object at the specified index will be reset/deleted after the list of ROIs was copied to the data pool. This avoids using the same ROIs a second time inadvertently. **Checkbox disabled:** After the copy has finished, the data object remains in the data tray at the specified index, so that you can reuse it.

Input-/Output data

Input	None.
Output	List of ROIs. A new data object is created in the data pool.

2.7.8.11.6 Copy Measurement List to Tray

2.7.8.11.6.1 Copy Measurement List to Tray: Introduction

Function

This function copies the measurement list from the data pool of the single check to the specified index of the global data-tray.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Destination index	Select here the zero-based index in the global tray where the measurement list will be stored. Index range: 0-99

Input-/Output data

Input	Measurement list.
Output	None.

2.7.8.11.7 Copy Measurement List from Tray

2.7.8.11.7.1 Copy Measurement List from Tray: Introduction

Function

The check function copies the measurement list at the specified index number from the global data tray to the data pool of a single check. The measurement list will then be available in the data pool as a new data object from type "Measurement list". The function will abort with an error, if the specified index does not contain a measurement list.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

This check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Copy from index No.	Selection of the zero-based index of the data tray, from where the measurement list will be copied. Index range: 0-99
Reset after copy operation	**Checkbox enabled:** If this checkbox is activated, then the object at the specified index will be reset/deleted after the measurement list was copied to the data pool. This avoids using the same measurement list a second time inadvertently. **Checkbox disabled:** After the copy has finished, the data object remains in the data tray at the specified index, so that you can reuse it.

Input-/Output data

Input	None.
Output	Measurement list. A new data object is created in the data pool.

2.7.8.11.8 Copy Histogram to Tray

2.7.8.11.8.1 Copy Histogram to Tray: Introduction

Function

This check function copies the histogram from the data pool of the single check to the specified index of the global data-tray.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Destination index	Select here the zero-based index in the global tray where the histogram will be stored. Index range: 0-99

Input-/Output data

Input	Histogram.
Output	None.

2.7.8.11.9 Copy Histogram from Tray

2.7.8.11.9.1 Copy Histogram from Tray: Introduction

Function

The check function copies the histogram at the specified index number from the global data tray to the data pool of a single check. The histogram will then be available in the data pool as a new data object from type "Histogram". The function will abort with an error, if the specified index does not contain a histogram.

Properties

Check function group Tools

The check function has parameters that can be changed dynamically.

This check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Copy from index No.	Selection of the zero-based index of the data tray, from where the histogram should be copied. Index range: 0-99
Reset after copy operation	**Checkbox enabled:** If this checkbox is activated, then the object at the specified index will be reset/deleted after the histogram was copied to the data pool. This avoids using the same histogram a second time inadvertently. **Checkbox disabled:** After the copy has finished, the data object remains in the data tray on the specified index, so that you can reuse it

Input-/Output data

Input	None.
Output	Histogram. A new data object is created in the data pool.

2.7.8.11.10 Control Tray

2.7.8.11.10.1 Control Tray: Introduction

Function

The check function controls the storage of objects in the Data Tray. Two modes are available: Data objects can be moved to a different index no. within the data tray (shift register functionality) or cleared from the index (to make sure they can only be read out once).

Properties

Check function group Tools

The check function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Operate on tray type	Sets the data type of the data tray on which the check function is to be applied. This can be done for all data types simultaneously or only for a certain

	selection. **All:** The method is applied to all data object types. **Image tray:** Select this check box to delete/move data objects of "Image" type. **ROIs tray:** Select this check box to delete/move data objects of "List of ROIs" type. **Measurement tray:** Select this check box to delete/move data objects of "Measurement list" type. **Histogram tray:** Select this check box to delete/move data objects of "Histogram" type.
Method	Select the mode to be used by the check function.
Shift elements	In shift mode, elements can be moved at will within the one-dimensional data tray in both directions (left and right). This mode can be used for implementing shift registers. Elements exceeding the index range of the data tray (0 - 99) because of the shifting are deleted. Empty elements are inserted at the shift starting point.
First element	Enter the index no. of the first data object in the data tray you want to shift.
Offset	Determines by how many places the objects are shifted. If you enter 0, no shift takes place.
Shift operation	Direction of shift. **Shift to the right**: Moves the contents towards higher index numbers. (With this operation, with an offset of 1, the content of index no. 0 is shifted to index no. 1.) **Shift to the left**: Moves the contents towards lower index numbers. (With this operation, with an offset of 1, the content of index no. 5 is shifted to index no. 4.)
Reset elements	Reset mode clears elements within the given range from the data tray. If the same index no. is entered in both fields **First element** and **Last element**, the range is reduced to a single element. If the index no. of the **First element** is bigger than the **Last element**, the range is inverted, i.e. all elements between **First element** and **Last element** remain, all others are deleted.
First element	Index no. of the first data object in the data tray to be deleted.
Last element	Index no. of the last data object in the data tray to be deleted.

Input-/Output data

Input	None
Output	None

2.7.9 Communication

2.7.9.1 Introduction to Check Function Group "Communication"

The check functions in group **Communication** can be used to communicate using digital I/O devices, field bus boards or other communication devices. They can be used without any object in the data pool. A configured data format converter is necessary to use these check functions.

2.7.9.2 Read Process Input Bit

2.7.9.2.1 Read Process Input Bit: Introduction

Function

This check function waits for a specified event occurring at a process input pin. If the event is registered within an adjustable time span, the check function reports correct execution, otherwise an error. This is a decision check function, i.e. the check may possibly be terminated due to the result of this function. Using the time-out mode, you can use this check function in three different manners.

Properties

Function group Communication

The function is a decision check function.

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Input	Selects the input pin to be read. The input pins are designated as in the data format converter.
Expected event	Selects, if the check function waits for the input pin to be "set" to **HIGH** or "reset" to **LOW**.
Time-out mode	Sets the manner the check function polls the input pin. **Read immediately, no time-out**: The input pin is read and evaluated once immediately. Strictly speaking, the function does not check for an event but for the state of the signal. **Maximum delay**: The check function reads the input pin repeatedly, and waits at most the time span given here for the required event to occur.

Wait indefinitely:
The check function reads the input pin repeatedly, and waits indefinitely long for the required event to occur.

Input / output data

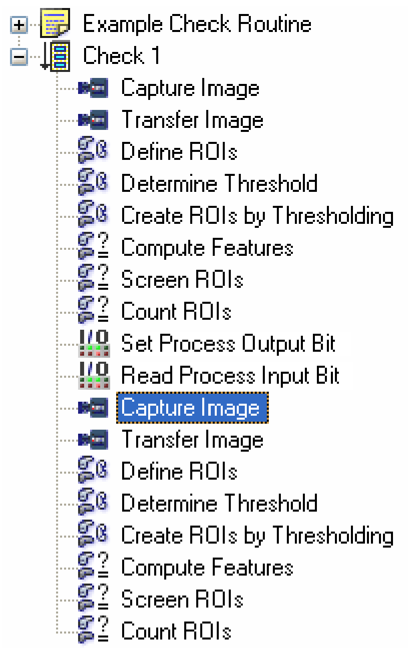
Input	None
Output	None

2.7.9.2.2 Read Process Input Bit: Examples

Check function **Read Process Input Bit** can be used together with check function **Set Process Output Bit** to configure simple communication protocols. It can also be used, when a test piece has to be moved during inspection.

The following check is configured for inspecting a large test piece with a single camera.

After the first part of the check, the test piece is moved in front of the camera for the second part to be inspected.



After completing the first part of the check, function **Set Digital Output** sets a digital output to HIGH. Check function **Read Process Input Bit** then waits until a certain input is set, before the next image is captured. The PLC can use this input to signal to NeuroCheck that the test piece has been repositioned. When a large and elastic test piece is repositioned, it may be necessary to wait until vibrations of the test piece have subsided before capturing the image. This can be done by inserting check function **Delay Execution** after check function **Read Process Input Bit** in order to stop the check for some time.

2.7.9.3 Set Process Output Bit

2.7.9.3.1 Set Process Output Bit: Introduction

Function

Sets the level of the process output pins of devices configured in the data format converter manager.

Properties

Function group Communication

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Tree view	The tree view lists all available output pins. The output pin designations correspond to those in the data format converter. After opening an output pin by double-clicking the output pins's name or by clicking the icon, two check boxes let you select whether the output pin is set to **HIGH** level or reset to **LOW** level. If both boxes are unchecked the current level of the output pin is unchanged.
Clear all settings	Unchecks all check boxes of all output pins, thus the check function will change no output level.
Set all output bits to reset state	Sets the settings of all output pins to the **Reset to LOW** state.

Input / output data

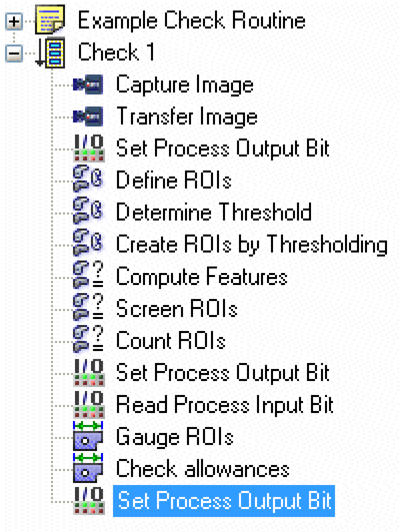
Input	None
Output	None

2.7.9.3.2 Set Process Output Bit: Examples

Check function **Set Process Output Bit** can be used together with check function **Read Process Input Bit** to construct simple communication protocols.

It is often necessary to enable a fast distinction between different types of faults on a test piece.

This can be done by configuring a check as follows:



At the beginning of the check a certain output is set to HIGH. Then objects are searched within the image, inspected and counted. If the number of objects does not correspond to the target number, check function **Count ROIs** yields "not OK", causing the check to terminate. The subsequent check function **Set Process Output Bit** will not be executed. It has been configured to reset the respective output to LOW, so the output will stay HIGH and a PLC can determine immediately from one of its digital inputs the type of error: not enough objects. If on the other hand the correct number of objects is present, the check goes on and a different output is set to HIGH. Check function **Check Allowances** may now yield "not OK" and terminate the check. Now the final instance of check function **Set Process Output Bit** will not be executed, the respective output stays HIGH and the PLC can determine that the correct number of objects has been present, but the objects did not comply with the required measurements.

2.7.9.4 Read Data into Register

2.7.9.4.1 Read Data into Register: Introduction

Function

This check function allows for fetching new data from a data format converter and writing the data into selected input registers at a determined moment during the check routine execution.

Properties

Function group Communication

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Available Registers	Select here the input registers that will be updated from the linked data format converter pin.

Input / output data

Input	None
Output	None

2.7.9.5 Send Data from Register

2.7.9.5.1 Send Data from Register: Introduction

Function

This check function allows for sending the data from selected output registers to a data format converter at a determined moment during the check routine execution.

Properties

Function group Communication

The function has a parameter dialog.

The parameter dialog contains the following elements:

Element	Description
Available Registers	Select here the output registers whose content will be sent to the linked data format converter pin.

Input / output data

Input	None
Output	None

2.7.10 Plug-In

2.7.10.1 Introduction to Check Function Group "Plug-In"

This check function group is only available in license level **Premium**. The check function group **Plug-In** may contain user defined Plug-In check functions.

Those check functions are not part of the NeuroCheck standard software, but were developed externally based on the specification of the NeuroCheck Plug-In interface. Plug-In check functions integrate seamlessly into the NeuroCheck development and runtime environment.

Check functions in this group are sorted in the order of the software extension modules (Plug-In-DLLs) they were implemented in, provided that the concerning DLL was loaded in the dialog **Software Settings ▸PlugIns**.

3. Hardware Integration

This section describes the integration of cameras and communication devices into the system, and additionally how to configure and manage these devices.

3.1 Hardware Integration: Introduction

An image processing system intended to perform automatic quality inspection depends on several hardware devices, some of them not found in ordinary PC systems. In substance these are cameras for image acquisition and communication devices required for the programs interaction with the manufacturing process.

The hardware configuration of NeuroCheck will change much more seldom than the actual inspection task. Hence the hardware configuration is stored as global software parameters valid for all check routines.

The core piece of hardware configuration is the NeuroCheck **Device Manager** dialog. From there the following hardware devices are configured:

- Security Key
- Cameras and frame grabbers
- Communication devices

3.2 Device Manager

The NeuroCheck device manager is the central dialog for managing all hardware components required for applying NeuroCheck as an industrial image processing system, integrated into automated manufacturing lines.

The dialog contains the following elements:

Element	Description
Tree view	This area displays each configured device, sorted by categories. Any device may be renamed here by selecting it and clicking it a second time to edit the name in-place. After selecting a device the buttons to the right and the context menu enable you to view and set device properties. The actual effect of the buttons depends on the selected device and is explained in the device specific sections. Devices currently unavailable are indicated by a symbol.
Device Info	This group displays in a table all device information of the selected device. Properties that are black instead of grey can be edited here.
About …	Opens a window that contains information about the currently used device driver of the selected item.
Help	Displays a help file, if available for this device.
New …	Choose this button to integrate a not yet configured device. The Hardware Wizard will aid you in the initial configuration of the device.
Properties …	Choose this button to view and set properties of the device selected in the tree view.
Reset!	Choose this button to reset the device selected in the tree view to its default settings.
Test …	Choose this button to test the device selected in the tree view.
Remove!	Choose this button to remove the device selected in the tree view. You will not be able to use this device with NeuroCheck without re-configuring it.
Import …	Choose this button to load a camera setup. (Available for devices of categories digital camera and frame grabber only.)
Export …	Choose this button to save the settings for the camera selected in the tree view in a separate file. (Camera settings are automatically stored in the global NeuroCheck configuration. The camera setup file is intended for transferring the settings to another NeuroCheck installation. Available for devices of categories digital camera and frame grabber only.)
Test Communication …	Opens a window to check the communication via the selected device. (Available for devices of categories digital I/O board and field bus board only.)

3.3 Add new device (Hardware Wizard)

The hardware wizard aids you in the initial configuration of one of the special hardware items required to use NeuroCheck for automatic visual inspections. It is called by choosing **New** in the Device Manager dialog.

> Please note that before including the device into the NeuroCheck device manager you need to install the vendor specific Windows device drivers for most device categories.

Select the hardware category

On the first page of the hardware wizard you have to select the type of hardware device to be configured from one of the following options:

- **Digital camera:**
  Digital cameras according to FireWire (IEEE 1394a/b) standard or Gigabit Ethernet cameras.
- **Frame grabber board:**
  If you want to run analog cameras or CameraLink cameras attached to a frame grabber board.
- **Digital I/O board:**
  Digital I/O boards.
- **Field bus board:**
  Boards for the communication via field bus networks, e. g. Profibus.
- **Industrial Ethernet:**
  For Ethernet based field bus communication (for instance Ethernet/IP, Profinet, EtherCAT).
- **Serial communication:**
  For simple stream based protocols (for instance RS232 or TCP/IP) run over a standard board.

Select the driver

On this page you can select the driver type from the list box. The software automatically scans for available drivers and provides them in this list. For some devices, a dialog can be opened by clicking the "Info" button. This dialog displays comprehensive information about the drivers required for the device and the drivers currently installed on your system.

> In addition to the drivers included in the standard software installation, NeuroCheck allows system integrators to add their own hardware to the system. Such drivers are indicated by a message from Hardware Wizard.

Select the device

On this page you can select the devices to be integrated. The list shows all devices according to the selected driver that are currently available and have not yet been integrated or are currently occupied.

As soon as you leave the wizard with **Finish** the device will be tested and integrated.

3.4 Managing Cameras

3.4.1 Managing Cameras

Camera configuration comprises several distinct layers:

- The camera setup, that means characteristics like exposure time, white balance or the selection of the trigger source. In a completed inspection station this setup will change comparatively seldom and is therefore stored in the global hardware configuration of NeuroCheck, configured through the Device Manager. Camera settings can be exchanged between different installations; see section "Camera setup files" for details.
- The adjustment of the cameras optic to environmental conditions, i.e. focusing and adjusting aperture. This requires continuous supervision of the image recorded by the camera and is therefore done in the Live view.
- Verification of the correct orientation of each camera with respect to the original configuration. See section "Reference images" for details.

Camera management therefore comprises the following steps:

- Configuring the camera:
  In the tree view of the Device Manager either select the digital camera or the camera input of the frame grabber the camera is connected to and choose **Properties**. It depends on the type of the digital camera or frame grabber board if a camera input configuration is available at all and which parameters can be set. The configuration dialog is therefore part of the device driver for the digital camera or frame grabber board and is explained in the documentation to the device driver. It is also possible to load ready-made camera configurations from a file. See section "Camera setup files" for more information.
- Renaming the camera:
  Since all functions using cameras refer to them by the names entered in the Device Manager dialog it may be convenient to change the names automatically assigned to the camera nodes during installation. A camera is renamed by selecting it in the tree view and clicking it a second time to edit the name in place.
- Choosing **Test** in the Device Manager dialog tests the selected camera.

The check routine accesses the configured cameras by using check functions of the **Image acquisition** group.

3.4.2 Camera Setup Files

The parameters set for each camera are considered as part of the overall setup of an inspection station and are therefore stored together with the other global program parameters of NeuroCheck. However, as these parameters describe the characteristics of specific camera models, which will be in use in different inspection stations, it is possible to store the camera parameters in separate files in order to transfer them to other systems or to transfer them to the replacement camera in case of a camera exchange.

How to store camera parameters

The characteristics of a specific camera are stored in a camera setup file by selecting the camera in the tree view of the Device Manager dialog and choosing **Export**. A file select dialog will appear, asking for the name of the file. This file will contain all parameters set for the camera (camera parameters are set from the Device Manager dialog by selecting the camera in the tree view and choosing **Properties**).

The file format and the file name extension depends on the camera and frame grabber driver respectively.

How to read camera parameters from a file

The parameters for a camera can be read from a camera setup file by selecting the camera node in the tree view of the Device Manager dialog and choosing **Import**. A file select dialog will appear, asking for the name of the file. The parameters of the selected camera will be set to the parameters stored in the file as explained above.

It is only possible to import a camera setup file that originally has been created with the same digital camera and frame grabber driver respectively the selected camera currently is connected to.

Using camera setup files

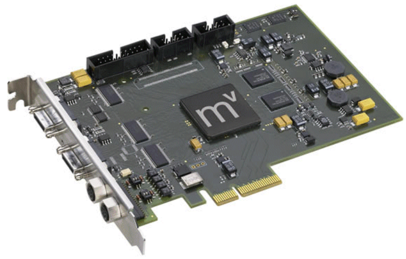
There are different uses for camera setup files:

- Document settings for specific camera types in human readable format by exporting the settings.
- Set identical parameters for different digital cameras and camera inputs of a frame grabber respectively: although it is perfectly possible to set all cameras to identical parameters by selecting each one in turn, choosing **Properties** and enter the same parameters over and over, this is very inconvenient and error prone. It is much easier to use setup files as follows:
    1. Configure one of the cameras and test the settings by choosing **Test**.
    2. Store the final parameters in a setup file by choosing **Export**.
    3. Select each of the other cameras in turn.
    4. For each one, choose **Import** and read in the newly created setup file.

- Transport camera settings from one inspection station to another:
    1. Select the camera whose settings are to be transferred.
    2. Choose **Export** to store the parameters in a setup file.
    3. Copy the newly created setup file to a data carrier and then from the data carrier to the hard disk of the other inspection computer (or transfer the file directly via network).
    4. In the NeuroCheck installation of the second inspection station, open the Device Manager, select the respective camera and choose **Import** to read in the copied setup file.
- Transport camera settings to a replacement camera in case of a defect:
    1. Select the camera whose settings are to be transferred in case of a defect.
    2. Choose **Export** to store the parameters in a setup file.
    3. In case of a defect exchange the camera.
    4. Select the replacement camera and choose **Import** to read in the saved setup file.

3.5 Managing Frame Grabbers

A frame grabber is used to transfer images from a camera that can't be connected with the PC by a standard PC interface to the PC memory.



Configuration of a frame grabber board comprises several aspects:

- The frame grabber has to be installed in NeuroCheck. This is done by choosing **New** in the Device Manager dialog and following the messages of the Hardware Wizard for installing a frame grabber board. After installation the board will appear in the tree view of the Device Manager with its camera inputs attached.
- The board has to be configured by choosing **Properties** in the Device Manager dialog when the frame grabber board has been selected in the tree view. The configuration dialog is part of the device driver for the board and is explained in the documentation belonging to the device driver. Availability of a configuration dialog depends on the device driver. It is possible to store the frame grabber parameters in separate files in order to transfer them to other systems or to transfer them to the replacement frame grabber in case of a frame grabber exchange.
- The camera inputs of the board have to be configured by selecting the respective input in the tree view and choosing **Properties**. It depends on the device driver if camera input configuration is available at all and which parameters can be set. The configuration dialog is therefore part of the device driver for the board. It is possible to store the camera parameters in separate files in order to transfer them to other systems or to transfer them to the replacement camera in case of a camera exchange.
- The camera inputs of the board may also be renamed. Since all functions using cameras refer to them by the names entered in the Device Manager dialog it may be convenient to change the names automatically assigned to the channels during installation of the board. A channel is renamed by selecting it in the tree view and clicking it a second time to edit the name in place.
- Choosing **Test** in the Device Manager dialog when the frame grabber board or one of its camera inputs has been selected in the tree view tests the device.

3.6 Managing Digital I/O

A digital I/O board can be used to remote-control NeuroCheck and to signal interior states to the controlling instance, e.g. a PLC.

Configuration of a digital I/O board comprises several aspects:

- The digital I/O board has to be installed in NeuroCheck; this is done by choosing **New** in the Device Manager dialog and following the messages of the Hardware Wizard for installing a digital I/O board. After installation the board will appear in the tree view of the Device Manager.
- The board has to be configured by choosing **Properties** in the Device Manager dialog when the digital I/O board has been selected in the tree view. The configuration dialog is part of the device driver for the board and is explained in the documentation to the device driver.
- In addition, a Data Format Converter must be set up for the digital I/O board in the Data Format Converter Manager. The Data Format Converter transforms the hardware bits of the digital I/O board to data that can be used by NeuroCheck and vice versa. The Data Format Converter can be set up automatically on closing the Device Manager or manually within the Data Format Converter Manager.
- The digital I/O board can be renamed by selecting it in the tree view and selecting it a second time to switch to the in-place editing mode.
- Choosing **Test communication ...** in the Device Manager dialog when the digital I/O board has been selected in the tree view calls the Test Digital I/O communication dialog.

3.7 Managing Field Bus Communication

A field bus board can be used to remote-control NeuroCheck and to signal interior states to the controlling instance, e.g. a PLC.

Configuration of a field bus board comprises several aspects:

- The field bus board has to be installed in NeuroCheck; this is done by choosing **New** in the Device Manager dialog and following the messages of the Hardware Wizard for installing a field bus board. After installation the board will appear in the tree view of the Device Manager.
- The board has to be configured by choosing **Properties** in the Device Manager dialog when the field bus board has been selected in the tree view. The configuration dialog is part of the device driver for the board and is explained in the documentation to the device driver.
- In addition, a Data Format Converter must be set up for the field bus board in the Data Format Converter Manager. The Data Format Converter transforms the bits of the field bus board to data that can be used by NeuroCheck and vice versa. The Data Format Converter can be set up automatically on closing the Device Manager or manually within the Data Format Converter Manager.
- The field bus board can be renamed by selecting it in the tree view and selecting it a second time to switch to the in-place editing mode.
- Choosing **Test communication ...** in the Device Manager dialog when the field bus board has been selected in the tree view calls the Test field bus communication dialog.

3.8 Managing Industrial Ethernet

An Industrial Ethernet communication can be used to remote-control NeuroCheck and to signal interior states to the controlling instance, e.g. a PLC.

Configuration of a Industrial Ethernet board comprises several aspects:

- The Industrial Ethernet board has to be installed in NeuroCheck; this is done by choosing **New** in the Device Manager dialog and following the messages of the Hardware Wizard for installing a Industrial Ethernet board. After installation the board will appear in the tree view of the Device Manager.
- The board has to be configured by choosing **Properties** in the Device Manager dialog when the Industrial Ethernet board has been selected in the tree view. The configuration dialog is part of the device driver for the board and is explained in the documentation to the device driver.
- In addition, a Data Format Converter must be set up for the Industrial Ethernet board in the Data Format Converter Manager. The Data Format Converter transforms the bits of the Industrial Ethernet board to data that can be used by NeuroCheck and vice versa. The Data Format Converter can be set up automatically on closing the Device Manager or manually within the Data Format Converter Manager.
- The Industrial Ethernet board can be renamed by selecting it in the tree view and selecting it a second time to switch to the in-place editing mode.
- Choosing **Test communication...** in the Device Manager dialog when the Industrial Ethernet board has been selected in the tree view calls the **Test Industrial Ethernet communication dialog**.

3.9 Managing Serial Communication

A serial interface or Ethernet connection can be used to remote-control NeuroCheck and to signal interior states to the controlling instance, e.g. a PLC. Due to the identical communication protocols, NeuroChecks manages the communication over serial interface as well as Ethernet with TCP/IP in the same category (please don't confound with Industrial Ethernet).

Configuration of the serial interface comprises several aspects:

- The serial interface and Ethernet connection respectively has to be installed in NeuroCheck; this is done by choosing **New** in the Device Manager dialog and following the messages of the Hardware Wizard for installing a serial interface and Ethernet connection respectively. After installation the interface or connection will appear in the tree view of the Device Manager with all its input and output channels attached.
- The interface or connection has to be configured by choosing **Properties** in the Device Manager dialog when the interface and connection respectively has been selected in the tree view. The configuration dialog is part of the device driver for the interface and connection respectively and is explained in the documentation to the device driver.
- In addition, a Data Format Converter must be set up for the interface and connection respectively in the Data Format Converter Manager. The Data Format Converter interprets the protocol data and extract the relevant data from the data stream and transforms them to data that can be used by NeuroCheck and vice versa. The Data Format Converter can be set up automatically on closing the Device Manager or manually within the Data Format Converter Manager.
- The serial interface and Ethernet connection can be renamed by selecting it in the tree view and selecting it a second time to switch to the in-place editing mode.
- Choosing **Test communication...** in the Device Manager dialog when the interface and connection respectively has been selected in the tree view calls the **Test serial communication** dialog.

4. Automatic Mode

Automatic mode is the operating mode for the automatic execution of check routines. This chapter describes how to configure this mode before and how to operate while it is running.

4.1 Introduction

4.1.1 Automatic Operation

In an automatic visual inspection station, the NeuroCheck software run in operating mode **Automatic**, fully integrated into the manufacturing line. In automatic mode, NeuroCheck is remote-controlled by an external signal source and sends result data to various destinations.

Automatic mode is started by choosing **Automatic** from the **Operating Modes** menu.

NeuroCheck then waits for a start signal from the source configured on the **Input Signals** page of the **Remote Control** dialog box to start an inspection run. It is also possible to change the current check routine automatically by another signal, configured in the same dialog box. After completion of the inspection run, NeuroCheck transmits the final result of the check routine (see "Check routine results" for an explanation of how this result is determined) to the destination configured on the **Output Signals** page of the **Remote Control** dialog box. In addition, result values of individual check functions can be transmitted or stored in a file (see "Result value output" for more information).

During inspection, NeuroCheck can display processing results in several output windows on the configurable automatic mode screen.

The following sections explain all the options for the configuration and operation in automatic mode in detail.

4.1.2 NeuroCheck on the Production Line

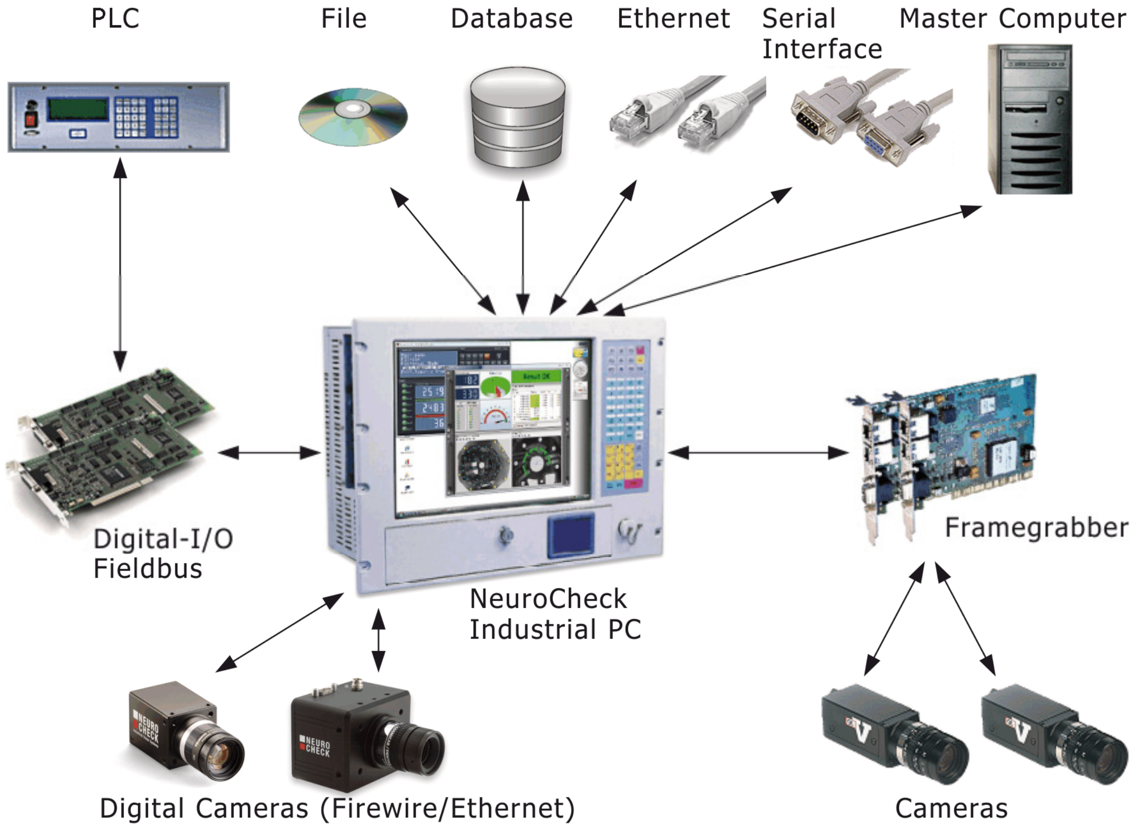
As can be seen from the figure below, a visual inspection system integrated into an automated production line has to support various hardware devices and communication channels, in order to:

- Facilitate remote-control of the vision system synchronized with the production process. This comprises for example starting an inspection run upon a specific signal, type-dependent switching of check routines, reporting completion and results of inspection runs etc. The devices and signals used for this purpose are described in section "Automatic Mode control".
- Transmit result data to various destinations, like PLCs, master computers etc. This topic is treated in depth in section "Data Output".
- Manage several cameras. More about this in section "Managing cameras".
- Give the operating personnel feedback on the current state of the inspection system. This is the task of the freely configurable Automatic mode screen.

- Manual interaction by the operating personnel. This possibility is treated in section "Manual Interaction".

These tasks in turn require hardware components not necessarily part of a standard PC installation. The **Device Manager** dialog provides a unified interface to the configuration of those hardware devices for use with NeuroCheck.

The following figure gives a schematic overview of a possible hardware setup for an automatic visual inspection system:

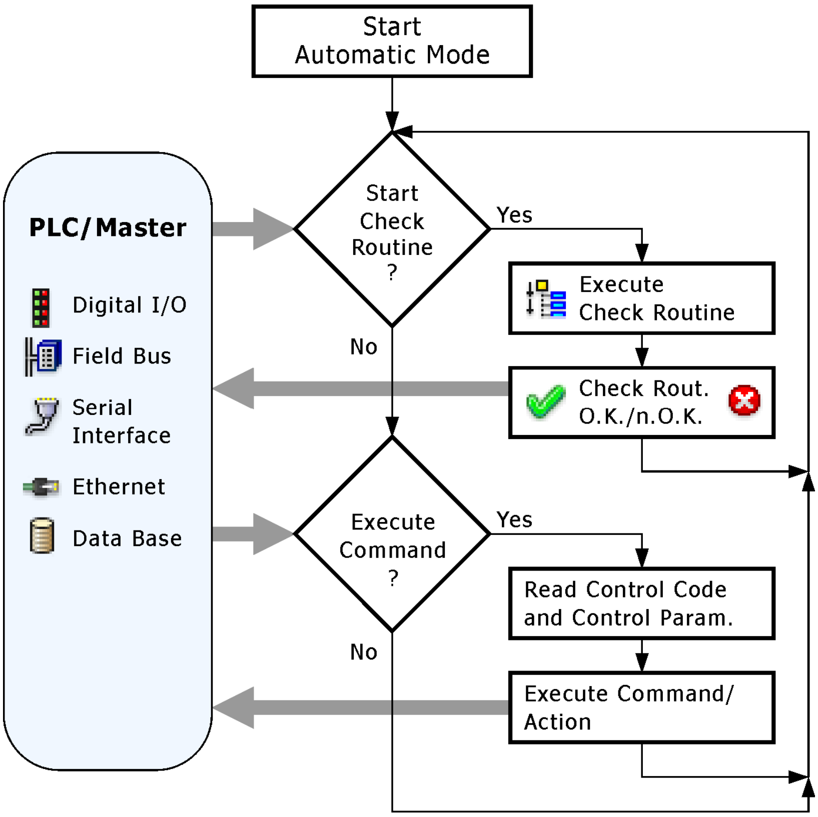


4.1.3 Automatic Inspection Procedure

The diagram depicts one run of an automatic inspection in NeuroCheck. After switching to automatic mode (see section "Start and Stop modes") NeuroCheck waits for a start signal from an external source, according to the settings on the **Input Signals** page of the **Remote Control** dialog box.

Upon receiving the start signal the check routine is executed. According to the settings made in the Data Output View, result data of individual check functions can be sent to various destinations for display and further processing.

After the check routine run has been completed, NeuroCheck determines a final check result, which can be either 'OK' or 'not O.K.' (see section "Check routine results" for an explanation of how this result is determined). This result can be transmitted to various destinations according to the settings on the **Output Signals** page of the **Remote Control** dialog box. This transmission is intended for process control purposes. In addition, the final result can also be sent to the same destinations as the results of individual check functions.



4.2 Automatic Screen

4.2.1 Configuration of Automatic Mode Screen: Introduction

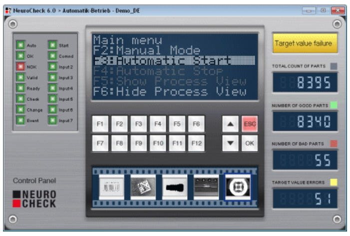
Usually, the automatic mode screen is divided into two windows:

- the **Control Panel** with the **Control Panel Menu**
- the **Process View**

When you are configuring a system and creating check routines, views for both windows are available by default. You can select a design from pre-defined views in the software settings for automatic mode (**System ▸ Software Settings ▸ Automatic Mode**).

Design and behavior of both windows can be freely configured in manual mode using graphical-interactive editors (so called "Designers").

Control Panel



The Control Panel visualizes the most important system information at a glance, however, details are not included.

Customized Control Panels are created using the Control Panel Designer and can be saved in the `Automatic Mode Configuration\Control Panels` program directory as `*.CPX file`.

For the user to be able to start actions manually or execute commands that are defined in the Control Panel Menu, at least one LCD panel, one CRT panel or alternatively functions keys should be available on the Control Panel.

After configuration, you have to integrate the Control Panel into NeuroCheck by assigning the appropriate `*.CPX file` to the system using the menu command **System ▸ Software Settings ▸ Automatic Mode ▸ Control Panel ▸ Design**!

Process View

Process View visualizes the process data created by the current check routine at run-time (e.g. images, measurement values).

User-defined process views are created using the Process View Designer and can be saved as `*.PVX files` in the `Automatic Mode Configuration\Process Views` program directory.

The process view file must be assigned to the check routine in the properties dialog! For this, right click on the check routine symbol in the structure view and start the **Check Routine Properties** ▸ **Automatic Mode** dialog from the context menu.

4.2.2 Process View Designer

4.2.2.1 Introduction to Process View Designer

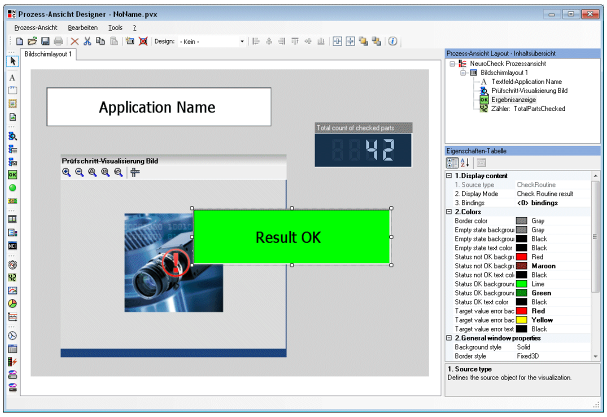
In automatic mode, the Process View window displays information about the running inspection process. Each check routine automatically uses a predefined Process View Layout that cannot be changed.

Using Process View Designer you can create an individual layout, if the standard layout does not meet the requirements of your inspection system. This "Designer" is a graphical-interactive editor that enables you to create your desired layout based on a plenty of predefined output windows. Within a Process View layout, you can even embed several screen layouts that can be switched in automatic mode during runtime.

When saving the Process View Layout, the draft is saved to the Standard Directory "`Configuration\Process Views`" as a *.PVX file. You can create and save any number of designs. The thus created Process View Layout can then be assigned to a check routine.

4.2.2.2 Process View Designer: Dialog

The Process View Layout for a check routine is freely configurable. A Process View Layout can contain several screen layouts, e.g. a standard layout for the running operation and a maintenance layout for diagnoses' purposes. In automatic mode you can switch between various screen layouts.



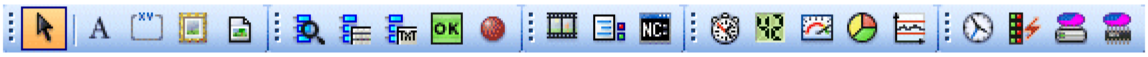
The Process View Designer is divided into three areas. The size of these areas can be adjusted using the mouse and drag and drop:

- Drawing area (left):
  The main area is the drawing area where the screen layout is configured. The drawing area contains a tab on the top containing the screen layout name. This allows for easy switching between drawing areas when you are using several screen layouts. Next to the drawing area, you'll find the menu and the toolbars (detailed description in the section Menu and Edit Toolbar and the Window Type Toolbar).
- List of windows (top right):
  The object list shows first the **Process View Layout**. All **Screen Layouts** are subordinated to the Process View Layout. The screen layouts contain all **Output windows** in the order they were added to the screen layout.
- Properties table (bottom right):
  When the Process View Layout, the Screen layout or an output window is selected from the drawing area or window list, this dynamic table displays the corresponding parameters (e.g. background color) in the categories (e.g. colors) in alphabetical order. The parameters of a category can be displayed or hidden using ⊟ or ⊞. For a parameter, the value can be changed directly using the keyboard (please note formatting) or using the or icon on the right border of the cell.
  Exception: a purely numeric field; this can only be changed via keyboard.
  If several output windows (on the drawing area) are selected, the table field contains only those options that can be changed for all simultaneously.

4.2.2.3 Create new Window in the Process View Designer

Choose window type

From the Window Type Toolbar to the left, select the window type you want to add to the process view. Alternatively, you can select the window type in the **Edit** ▸**Add Window** menu. A list of all available window types can be found in chapter Window Type Toolbar.



Add window to process view layout

The window is added to the screen layout by selecting the displayed icon from the window type toolbar. Set the starting point by clicking the left mouse button. Hold the left mouse button down and drag the mouse to the desired location at the end of the box. A thick line indicates the current proportions, a thin line the line-out for alignment with the next object. The window will be created by releasing the left mouse button.

> Some window types can only be inserted once into a screen layout and they are then disabled (gray) in the window type toolbar.

Select and edit an output window

After a window has been drawn, Selection Mode is automatically activated. Position and size of most output windows can be edited in the drawing area.

Edit function and layout of output window

- Edit the functionality and layout of the new output window in the properties table on the bottom right. The properties can vary greatly depending on the selected window type.
- The most important settings are always in the uppermost category whose title begins with "1.". For example, for many window types you can here set the links to the check routine or system objects whose contents will be displayed in the window.
- Usually, you can also determine the position, size and colors of a window in the properties table.
- For some output window types special options are available that are listed in the last category of the properties table, **Specific Window Properties**.

4.2.2.4 Skins

Skins help you to create appealing layouts quickly. Use the list box in the menu toolbar to assign the selected graphical design (skin) to all windows of the current layout.

4.2.2.5 Window types

Here you can find windows that can be added to the Process View. Most window types can be used several times, but some only once per Process View.

The following window types are available for the Process View Layout:

- Text label
- Group frame
- Static image
- File viewer
- Check function image
- Check function list
- Check function message
- Result status
- LED
- Thumbnail image strip
- System log
- Trace messages
- Execution time
- Counter
- Result rate tachometer
- Chart
- Measurement history
- Date / time
- Data register viewer
- Digital I/O or field bus state
- Hard disc usage
- Memory usage

4.2.2.6 Embedding Process View Layout into NeuroCheck

NeuroCheck automatically uses a predefined standardized process view layout for each check routine.

There are two ways to use a user-defined process view layout in automatic mode:

- Assign the user-defined process view layout directly to a check routine.
- Define the user-defined process view layout as the new standard for the system.

Assigning the user-defined process view layout directly to a check routine.

Open the **check routine properties** dialog by right clicking the check routine in the structure view. Select **Properties** from the context menu. The **Check Routine Properties** dialog is opened. Select **Automatic Mode ▸Process View**. Here you can assign an individual process view layout to your check routine.

Defining the user-defined process view layout as the new standard for the system.

Open the **Software Settings** dialog in the **System** main menu. Go to page **Automatic Mode ▸ Process View ▸Design**. Here you can enter your individual process view layout as the new standard.

4.2.3 Control Panel Designer

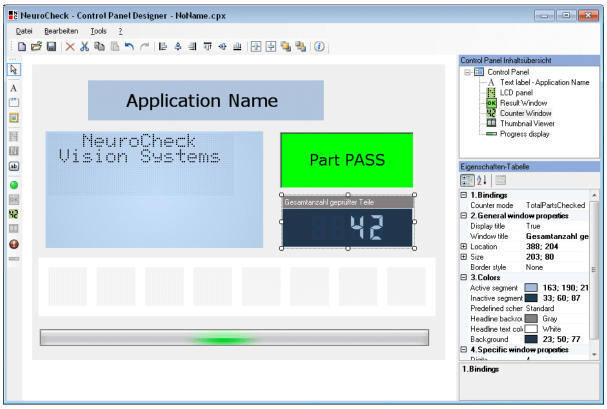
4.2.3.1 Introduction to Control Panel Designer

The Control Panel serves for entering control commands and displays the most important system information. It is possible to use predefined Control Panels. These cannot be changed.

If the pre-defined Control Panels do not satisfy the requirements of your inspection system, you can create individual Control Panels using the Control Panel Designer. This "Designer" is a graphical-interactive editor that enables you to create your desired layout based on predefined windows (objects).

When saving the Control Panel, the draft is saved in the standard directory `"Configuration\Control Panels"` as a `*.CPX file`. You can create and save any number of designs.

4.2.3.2 Control Panel Designer: Dialog



The Control Panel Designer is divided into three areas. The size of these areas can be adjusted using the mouse and drag and drop:

- Image panel (left):
  The main area is the drawing area where the Control Panel is configured. Next to it, you'll find the menu and toolbars (detailed description in section Menu and Edit Toolbar and Window Type Toolbar).
- List of windows (right upper corner):
  This list contains all windows in the order in which they were added to the Control Panel.
- Properties table (right lower corner):
  When the Control Panel or a window (in image panel or object list) is selected, this dynamic table field displays the corresponding parameters (e.g. background color) in the categories (e.g. colors) in alphabetical order. The parameters of a category can be displayed or hidden using ⊟ or ⊞. For a parameter, the value can be changed directly using the keyboard (please note formatting) or using the ⌄ or ... icon on the right border of the cell. Exception: a purely numeric field; this can only be changed via keyboard.
  If several windows (on the drawing area) are selected, the table field contains only those options that can be changed for all simultaneously.

4.2.3.3 Create new Window in the Control Panel Designer

Choose window type

From the Window type toolbar to the left, select the window type you want to add to the Control Panel. A list of all available window types can be found in chapter Window type toolbar.

Add window to the Control Panel

The window is added to Control Panel by selecting the displayed icon from the window type toolbar. Set the starting point by clicking the left mouse button. Hold the left mouse button and drag the mouse to the desired location of the end of the window. A thick line indicates the current proportions, a thin line the line-out for alignment with the next window. The window will be created by releasing the left mouse button.

> Some elements of the object toolbar can only be inserted once and they are then disabled (gray) in the object toolbar.

Select and edit window

 After a window has been drawn, Selection Mode is activated automatically. Position and size of most windows can be edited in the drawing area.

Edit function and layout of windows

- Edit the functionality and layout of the new output window in the properties table on the bottom right. The properties can vary greatly depending on the selected window type.
- The most important settings are always in the uppermost category whose title begins with "1.". For example, for many window types you can here set the links to the system object whose contents will be displayed in the window.
- Usually, you can also determine the position, size and colors of a window in the properties table.
- For some window types special options are available that are listed in the last category of the properties table, **Specific Window Properties**.

4.2.3.4 Window types

4.2.3.4.1 Control Panel Window Type Toolbar

Here you can find windows that can be added to the Control Panel. Most window types can be used several times, but some only once per Control Panel.

The following window types are available for the Control Panel:

- Text label
- Group box
- Bitmap
- LCD matrix display
- CRT display
- Button
- LED
- Result status message display
- Counter
- Thumbnail image list
- Last Errors
- Progress bar

4.2.3.4.2 Window Type Text Label

A text box displays static or dynamic information in text form. Static text can be used, for example, for additional labeling of other windows in the Control Panel. Dynamic text can display current system information in a compact fashion.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.3 Window Type Group Box

To enhance clarity, you can group windows in a group box (e.g. arrangement of all LEDs for digital inputs).

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.4 Window Type Bitmap

Creates an area where an image can be placed (e.g. company logo).

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.5 Window Type LCD Matrix

Creates an area to act as a placeholder for the Control Panel Menu. The design looks like a LCD display ("Liquid Crystal Display"). This object can only be added once. Content and function of the command structure are not configured here but in the Control Panel Menu Editor. There, you also have to configure the following:

- the content of the LCD display and
- what function key (e.g. F1) causes which interaction in the process.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.6 Window Type CRT display

Creates an area to act as a placeholder for the Control Panel Menu. The design looks like a CRT display ("Cathode Ray Tube"). This object can only be added once. Content and function of the command structure are not configured here but in the Control Panel Menu Editor. There, you also have to configure the following:

- the content of the CRT display and
- what function key (e.g. F1) causes which interaction in the process.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.7 Window Type Button

Creates a button. This button can be linked to a function key so that a command is executed upon selecting the button in automatic mode. The links between key and command depend on the settings in the Control Panel Menu.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.8 Window Type LED

Creates a LED ("Light-emitting diode"). The LED shows the status, for example, of a digital input.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.9 Window Type Result Status Message Display

Creates a box displaying the final result of the inspection. This window can only be added once.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

You set the textual content of the result display on the **Result Output** page of the **Check Routine Properties Dialog**!

4.2.3.4.10 Window Type Counter

Creates a counter window. In this window you can, e.g., display the total number of inspected parts, the number of parts per minute or the number of NOK parts.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.11 Window Type Thumbnail List

Creates a box displaying the last images captured in the current inspection process. This window can only be added once.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.4.12 Window Type Progress Bar

Creates a box containing a progress bar. In automatic mode, the progress bar is only displayed during a waiting period. This window can only be added once.

Select the window in the window list or in the drawing area of the Control Panel to edit the window properties in the properties table.

4.2.3.5 Embedding control panel into NeuroCheck

To be able to use a Control Panel in automatic mode, it must first be defined in the software settings. Only a single Control Panel can be used per system. Open the **Software Settings** dialog in the **System ▸ Software Settings** main menu.

- Using predefined Control Panels:
  NeuroCheck supplies a number of predefined Control Panels. You'll find this selection in the **Software Settings ▸Automatic Mode ▸Control Panels** dialog.
- Using user-defined Control Panels:
  To use one of the user-defined Control Panels, open the **Software Settings ▸Automatic Mode ▸ Control Panels** dialog and select **Custom Design…**. Here you can import your individual Control Panel.

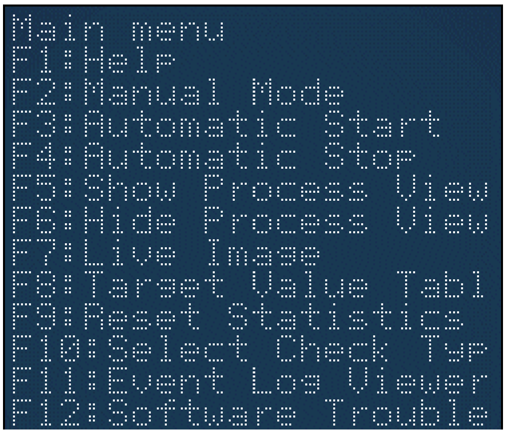
4.3 Automatic Mode Interaction

4.3.1 Introduction

Even a system for automatic inspection operation requires a certain amount of manual interaction. This chapter describes how to configure these interactions in advance.

- The Control Panel Menu defines the structure for potential commands that can be called manually using the function keys of the keyboard in automatic mode. You can create a function key structure using the Control Panel Menu Editor.
- Graphical buttons linked to Function keys can be integrated into the Control Panel.
- For the Process View, you can configure ways for zooming and scrolling.
- Password-protected security profiles specify access to functions individually.

4.3.2 Control Panel Menu



The Control Panel Menu defines the actions or commands the user can execute in automatic mode and their assignment to the function keys.

This function key structure for commands and sub-menus can be presented to the user in the Control Panel as a menu view or as function key buttons for operation via keyboard or mouse.

If you are configuring a system, a default function key structure is available. You can select from pre-defined function key structures in software settings (**System ▸ Software Settings ▸ Automatic Mode ▸ Control Panel**).

If none of the pre-defined Control Panel Menu satisfies the requirements of your inspection system, you can create individual Control Panel Menus using the Control Panel Menu Editor.

If the system requires more than the usual 12 function keys, the commands can be structured in sub-menus. Please note that there might be a problem with documenting the multiple functions of the function keys.

4.3.3 Control Panel Menu Editor

4.3.3.1 Introduction to Control Panel Menu Editor

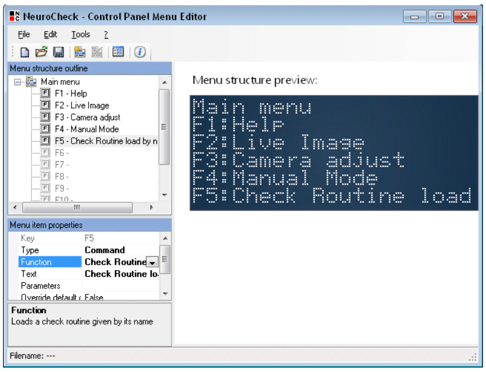
The Control Panel uses a function key structure that can be configured using the **Control Panel Menu Editor**. The Control Panel Menu defines the actions or commands the user can execute in automatic mode and their assignment to the function keys.

You can use a pre-defined Control Panel Menu. This cannot be changed. If the pre-defined Control Panel Menu does not satisfy the requirements of your inspection system, you can create individual Control Panel Menus using the Control Panel Menu Editor. If the system requires more than the usual 12 function keys, the commands can be structured in sub-menus.

The **Control Panel Menu Editor** saves all designs as `*.MNUX files` in the `Automatic Mode Configuration\Control Panel Menus` directory. You can create and save any number of designs.

After configuration, you have to integrate the Control Panel Menu into NeuroCheck by assigning the appropriate `*.MNUX` file to the system using the menu command **System** ▸ **Software Settings** ▸ **Automatic Mode** ▸ **Control Panel** ▸ **Menu**!

4.3.3.2 Control Panel Menu Editor: Dialog



The window of the Control Panel Menu Editor is divided into three areas whose size can be adjusted with the mouse:

Function key structure for commands and sub-menus (top left)

This overview shows the function key structure where you can define commands and sub-menus. First, it contains the **main menu** with the **function keys F1 - F12**. Every function key is linked either to a command, a sub-menu or nothing at all. In turn, every sub-menu contains the **function keys F1 - F12**.

- **Add command:**
  To link a command you have to select a function key in the function key structure overview on the left. On the bottom, change the **type** property to **command**. In **Function**, select the desired command. For some commands, it is necessary to enter a **parameter**.
- **Add sub-menu:**
  To add a sub-menu, you have to select a function key in the function key structure overview on the left. Use **Ctrl+M** or the icon in the tool bar to add the sub-menu. Alternatively, you can change the **type** property to **sub-menu** on the bottom.

Properties table (bottom left)

If you select a function key in the top left structure overview, this dynamic table field displays the corresponding properties. For each property of this function key, the value can be changed directly via keyboard (please note formatting), for some fields also using the icon on the cell's right border.

> If a function already linked to a command is changed into a sub-menu, all further settings of the command will be lost.

Preview (on the right)

Here a permanent preview of what the menu and command structure defined on the left would look like on the Control Panel is displayed, which is currently defined in Software Settings.

4.3.3.3 Embedding a Control Panel Menu into NeuroCheck

To be able to use a Control Panel Menu in automatic mode, it must first be defined in the software settings. Only one Control Panel Menu can be used per system. Open the **Software Settings** dialog in the **System ▸ Software Settings** main menu.

Use predefined Control Panel Menu

NeuroCheck offers several predefined Control Panel Menus to choose from. You'll find these in the **Software Settings ▸Automatic Mode ▸Control Panel ▸Menu** dialog.

Use customized Control Panel Menu

To use one of the customized Control Panel Menus, open the **Software Settings ▸Automatic Mode ▸ Control Panel ▸Menu** dialog and select **Individual Design**. You can import your individual Control Panel.

4.3.4 Function key buttons

Besides displaying the function key structure of the Control Panel Menu as a menu, you can use graphical function key buttons in the Control Panel. These buttons can be created using the Control Panel Designer and furnished with text or a background image. Clicking on such a button executes the command linked to the function key in the Control Panel Menu command structure.

> Please note that there might be a problem with multiple functions of the function keys when using sub-menus.

4.3.5 Interaction in process view

Use the **System** ▸ **Software Settings** ▸ **Automatic Mode** ▸ **Process View** ▸ **Interaction** menu to configure to what extent the user can interact in the process view window with the mouse during runtime.

For example, the potential to zoom and scroll in images can be helpful and interesting in many cases, but it can cause a slow-down of the inspection process.

4.3.6 Interaction using security profiles

Password-protected security profiles allow for individually limited access to functions. For more details see Security Profiles.

> To change user profiles in automatic mode, activate the appropriate command on the Control Panel Menu.

4.4 Automatic Mode Control

4.4.1 Introduction

In automatic mode NeuroCheck can be remote-controlled by external sources (i.e. via digital I/O or serial interface). Options for controlling automatic operation are set in the Remote Control dialog box and include:

- Start of automatic checking after a signal received via digital I/O, field bus, serial interface or custom communication, as well as for simulation purposes upon key press or in regular intervals.
- Automatic change of check routine in automatic mode, triggered by a signal on digital I/O, field bus, serial interface or custom communication.
- Execution of further commands upon request by digital signals.
- Report of final check result via digital I/O, field bus, serial interface or custom communication.
- Report of several system states.

For control of automatic operation via digital I/O, field bus, serial interface or custom communication the respective interface has to be configured in the Device Manager dialog. Furthermore a Data Format Converter has to be configured for the appropriate device.

The assignment of the above mentioned signals to the input and output channels is set in the Input Signals and Output Signals pages of the **Remote Control** dialog box.

It is easily possible to control NeuroCheck completely via digital I/O and **in addition** sent the final result via serial interface (e.g. when the control is performed by a PLC, using digital signals for communication with NeuroCheck , whereas a master computer wants to receive the result via a serial channel). More about this in section "Result value output".

4.4.2 Remote Control Dialog box

The main concern of automatic mode configuration is remote-control of NeuroCheck by external sources during automatic operation, i.e. starting, stopping and switching check routines etc. All these settings are made in this **Remote Control** dialog box, opened by choosing **Remote control** from the System menu. The dialog box has two pages for the different aspects of remote-control in automatic mode.

- "Input signals"
  used to start the inspection run and to remote-control by several commands.
- "Output signals"
  used to signal several system states like the completion of an inspection run and to report final results.

4.4.3 Input Signals

4.4.3.1 Input Signals: Introduction

The property page **Input Signals** of the **Remote Control** dialog box enables you to specify NeuroCheck's reaction to signals in automatic mode.

NeuroCheck understands the following signals in automatic mode:

- **Start check routine**:
  After receiving this signal an inspection run using the current check routine is started.
- **Execute command**:
  After receiving this signal NeuroCheck checks the data currently present on the input pins **Command Code** and **Command Parameter**. Dependant upon those it executes the requested command. Thus you always have to connect all those three rows to input pins if you want to enable command execution.

The table contains the following elements:

Element	Description
Function of signal	This column lists the signals available to remote-control NeuroCheck in automatic mode. If the check box in front of a signal name is activated, NeuroCheck will react to the signal coming from the source displayed in the last column.
Data type	Data type of the input signal (signal or ID)
Source [for signal "Start check routine"]	Opens the **Select Start Signal Source** dialog to specify the source for the start signal.
Source [for signal "Execute command"]	Opens the **Connect Process Signal** dialog to specify the source for the signal selected in the input signal list.

4.4.3.2 List of Input Commands

NeuroCheck can execute the following commands, if you have enabled this feature on property page **Input Signals** of the **Remote Control** dialog box:

Name	**Command Code (integer)**	**Command Code (Binary encoded)**	**Command Parameter**	**Behaviour**
Reset last command acknowledgement	0	0	-	After receiving this command NeuroCheck resets its output signal **Last Command** to 0.
Select check routine	1	1	New check routine ID	After receiving this command NeuroCheck tries to load the check routine with the identification number specified in the **Command Parameter**. This depends on the settings on the **Check Type Selection** page of the **Software Settings ▸ Automatic Mode** dialog box. The check routine will be looked for either in the directory specified there or in the list configured there.
Select security profile	2	10	New security profile ID	After receiving this command NeuroCheck changes the current security profile to the one specified in the **Command Parameter**. This depends on the settings in the security profiles dialog box. If there is no valid profile ID present, then the system switches to the default profile which has ID zero by definition.
Reset statistics	3	11	-	After receiving this command NeuroCheck will reset the statistic counters for the currently loaded check routine which are displayed in the Control Panel and Process View. This signal has the same effect as command Reset statistics from the **Tools** menu.
Adjust camera start	4	100	Camera index	After receiving this command NeuroCheck will display the Adjust Cameras dialog for the comparison of reference images with current live

				camera images. You can direct this to a certain camera by using the **Command Parameter**.
Adjust camera end	5	101	-	After receiving this command NeuroCheck will close the Adjust Cameras dialog.
Self test start	6	110	-	After receiving this command NeuroCheck will open the self-test dialog and perform the self-test.
Self test end	7	111	-	After receiving this command NeuroCheck will close the self-test dialog disregarding its result.
Simulate function key	8	1000	0 = No action 1 = F1 2 = F2 3 = F3 4 = F4 5 = F5 6 = F6 7 = F7 8 = F8 9 = F9 10 = F10 11 = F11 12 = F12 13 = Escape 14 = Arrow up 15 = Arrow down 16 = Enter 17 = Arrow left 18 = Arrow right	After receiving this command NeuroCheck will try to execute the action of the current level of the Control Panel Menu that has been assigned the same function key ID as specified in the **Command Parameter**. In addition to the function key ID some internal actions are also available.
Automatic mode terminate	9	1001	-	After receiving this command NeuroCheck will abort any running check routine execution immediately and switch to the "Automatic stop" state.

				Please note that this signal will not be recognized if the input signal **Start check routine** is HIGH at the same time. Please also read the warning note in section "Terminate running Automatic Mode".
Shutdown	10	1010	-	After receiving this command NeuroCheck will try to close itself, and to cause a computer shutdown by the operating system. Please note the options for this command available in the Software Settings dialog.
Select process view screen layout	11	1011	New screen layout ID	After receiving this command NeuroCheck will change the process view screen layout to the one specified in the **Command Parameter**.
Check Routine terminate	12	1100	-	After receiving this command NeuroCheck will abort any running check routine execution immediately. The state "Automatic started" will remain though. Please note that this signal will not be recognized if the input signal **Start check routine** is HIGH at the same time. Please also read the warning note in section "Terminate running Automatic Mode".
Register Reset	13	1101	0 = All 1 = Input register 2 = Output register 3 = Free register	After receiving this command NeuroCheck resets the data registers to default values. You can restrict this to a certain register category by using the **Command Parameter**.
File output series switch	14	1110	**All bits set**: Notify all DFC. **Else**: 4 low-level	After receiving this command NeuroCheck will try to notify the data format converters. This is meant for data format converters of the **File** category. You can restrict this notification to a certain data format

			bits = Index of DFC assembly within all DFC assemblies of type "file". 4 high-level bits = Channel ID of file within this DFC.	converter by using the **Command Parameter** (8-bit only)

Please note that the range available for the transmission of the **Command Code** and **Command Parameter** values may be limited by the implementation of the Data Format Converter. For instance, when using the NeuroCheck standard Data Format Converter for the device type **Digital-I/O** only 4 Bits each are allocated for this, so you can only transmit values ranging from 0–15.

4.4.4 Output Signals

4.4.4.1 Output Signals: Introduction

The property page **Output Signals** of the **Remote Control** dialog box enables you to specify the signals sent by NeuroCheck in automatic mode.

The table contains the following elements:

Element	Description
Function of signal	This column lists the signals which can be sent by NeuroCheck in automatic mode. If the check box in front of a signal name is activated, NeuroCheck will sent this signal to the destination displayed in the last column.
Data type	Data type of the output signal (signal or ID)
Signal destination	Opens the Connect Process Signal dialog to specify the destination for the signal selected in the output signal list.

4.4.4.2 List of Output Signals

The property page **Output Signals** of the **Remote Control** dialog box enables you to specify the signals sent by NeuroCheck in automatic mode.

NeuroCheck generates the following signals in automatic mode:

- **System alive**:
  The selected output is set to HIGH immediately after program start. It is set to LOW just before the NeuroCheck is closed. Therefore, the signal is HIGH as long as the NeuroCheck software is running.
- **Automatic mode enabled**:
  This output indicates that the software is in Automatic Mode and that Automatic Mode is in the **Automatic Start** state.
- **System ready**:
  In automatic mode, this output is set to HIGH immediately after an inspection run has been finished or after an execute command operation has been finished. It is set to LOW after a start signal for a new inspection run or an execute command signal has been received..
- **Data valid**:
  If this output is HIGH, then the check routine results can be treated as up-to-date and valid.
- **Result OK**:
  This signal is sent after an inspection run has been finished. It is HIGH if the check routine result is "OK".
- **Result not OK**:
  This signal is sent after an inspection run has been finished. It is HIGH if the check routine result is "not OK".
- **Check Routine changed**:
  This signal indicates that the current check routine has been changed since the last saving of the check routine file.
- **Self test failed**:
  This signal indicates that a kicked-off self-test has been processed with the result "not OK".
- **Event log full**:
  This signal indicates that the internal event log can not be carried on due to too much data amount.
- **Current check routine ID**:
  Mirror of the ID of the currently loaded check routine.
- **Current security profile ID**:
  Mirror of the currently logged in NeuroCheck security profile ID.
- **Last command code**:
  Display the command code of the last action that has been executed successfully after receipt of an **Execute command** signal.

4.4.5 Signal types

Signals from the viewpoint of the Data Format Converters

The architectural layer of the Data Format Converters abstracts the physical properties of the external signals completely isolating the NeuroCheck software from this level. How the signals are communicated externally is therefore part of the implementation of the Data Format Converter used. Depending on the communication medium used, the transfer of the signals on the side of the periphery can be done in very different ways: parallel (e.g. digital I/O) or serial communication, signal-edge-based or level-based signal interpretation and much more. The way the Data Format Converter treats external signals can therefore only be described in the documentation of the Data format Converter used.

Signals from the viewpoint of NeuroCheck

NeuroCheck differentiates between the following signal types:

- input signals that can be described with one information unit (1 bit)
  Examples: input signals "Start Check Routine" and "Execute Command". NeuroCheck polls these input signals from the Data Format Converters. In the case of digital I/Os, for example, this means that NeuroCheck does not react to a rising edge but the state of the input.
- Input signals requiring several information units (n bits).
  Examples: input signals "Command Code" and "Command Parameter". NeuroCheck polls these input signals from the Data Format Converters. It's up to the implementation of the Data Format Converter to have the full information asked for at this time.
- Output signals that can be described with one information unit (1 bit)
  Example: output signal "System ready". NeuroCheck indicates a state with these output signals existing over a period of time. Upon entry into the state, NeuroCheck sends the signal to the Data Format Converter once, and sends the inverted signal once at leaving the state. It's up to the implementation of the Data Format Converter to process the information in the appropriate way for the output medium.
- Output signals requiring several information units (n bits).
  Examples: Output signals "Current check routine ID" and "Current security profile ID". NeuroCheck treats these output signals in an identical way to the 1-bit output signals.

4.4.6 Digital Communication in Automatic Mode

This section describes the sequence of signals sent and expected by NeuroCheck during automatic inspection run, when a communication device has been configured in the Device Manager dialog.

> All descriptions in this section assume that the appropriate signals have been activated on page **Input Signals** resp. page **Output Signals** of the **Remote Control** dialog box.
>
> Please also note the section Signal Types.

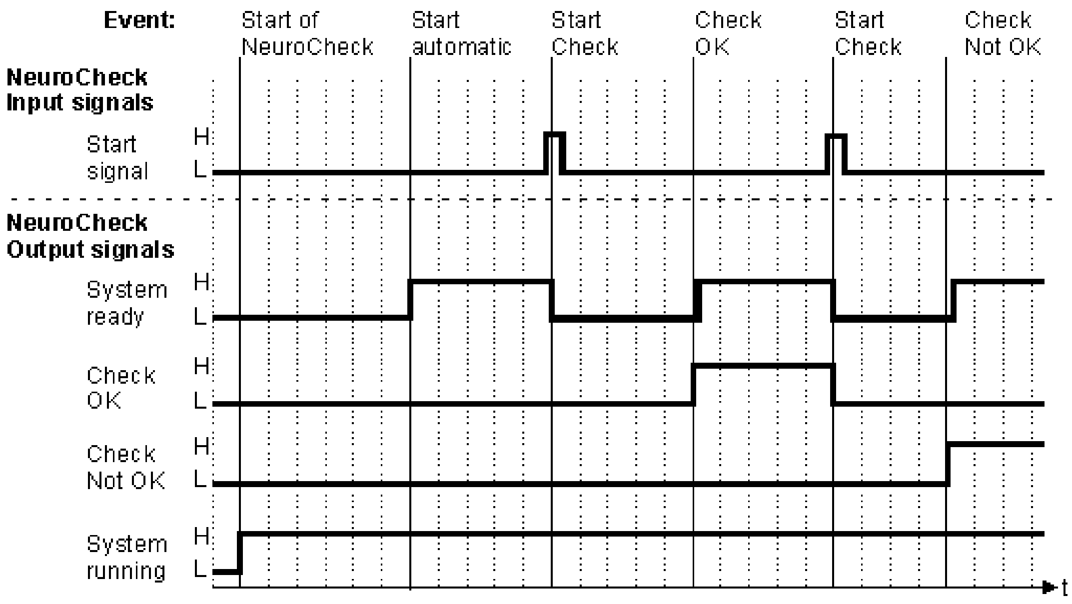
- In Manual Mode or in Automatic Mode state **Automatic Stop** NeuroCheck does not listen to input signals. Only the output signal **System alive** is set.
- Upon switching automatic mode into the **Automatic Start** state, NeuroCheck also sets the **Automatic mode enabled** and **System ready** signals.
- NeuroCheck is waiting in a loop whether one of the configured Input Signals will be set.
- As soon as the input signal **Start check routine** is received from the configured source, NeuroCheck reacts as follows:
    - These output signals are reset: **System ready**, **Result OK** and **Result not OK** (maybe still set from the previous inspection run), **Data valid**.
    - NeuroCheck executes the inspection run using the currently active check routine.
    - After the inspection run has been completed, NeuroCheck sets both final result signals to the appropriate state: **Result OK** and **Result not OK**.
    - Finally NeuroCheck sets the output signals **Data valid** and **System ready** indicating that the results can now be used.
- As soon as the input signal **Execute command** is received, NeuroCheck reacts as follows:
    - NeuroCheck checks the input signals **Command Code** and **Command Parameter**.
    - The output signal **Last command** is reset.
    - The output signal **System ready** is reset.
      Please note: For actions that are executed within a very short period of time, you may not be able to detect the reset of this output signal.
    - Depending on these values, NeuroCheck will try to execute the command.
    - After completing the action - irrespective of the result - NeuroCheck sets the output signal **System ready**.
    - On successfully completing the action, NeuroCheck sets the output signal **Last Command** to the **Command Code** used.
- The **Automatic Terminate** and **Check Routine Terminate** commands receive a special treatment. During the check routine execution, these signals are checked periodically in a separate background process, unless the signal **Start check routine** is set at the same time.

More information:

- **Signal flow for start pulses**

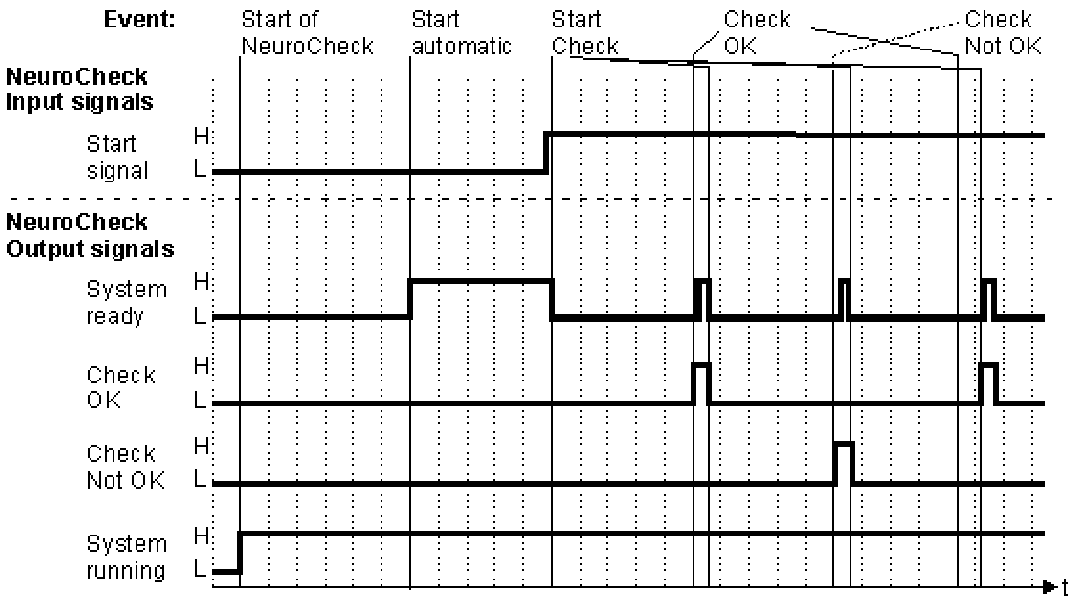
  The following diagram shows the signal flow when short pulses are used as start signals. Please note that

  - the start signal should stay HIGH at least until NeuroCheck has reset the **System ready** signal in order to guarantee correct hand-shaking,
  - the result signals **Result OK**, **Result not OK** are set before the **System ready** signal (the time span is exaggerated in the diagram; there is no actual delay, it is simply the sequence in which the signals are set).



- **Signal flow for static start signal**

  You can achieve an extremely fast inspection sequence by setting the start signal statically HIGH. Since NeuroCheck checks the start signal for level instead of edge it will recognize the start signal again immediately after an inspection run has ended. The width of the **System ready** signal in the diagram is exaggerated, it will be HIGH only for the time NeuroCheck needs to read the start signal again. This speed can be set on the **Remote Control Options** page of the **Software Settings** dialog box.



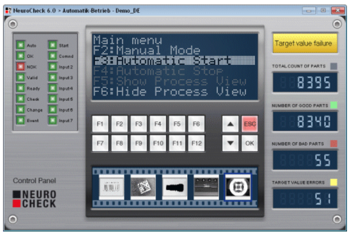
4.5 Operating the Automatic Mode

4.5.1 Operating the Automatic Mode: Overview

This chapter describes how to operate NeuroCheck automatic mode.

Usually, the NeuroCheck automatic mode screen is divided into two windows:

The **Control Panel** window hosting the **Control Panel Menu**:



The **Process View** window:

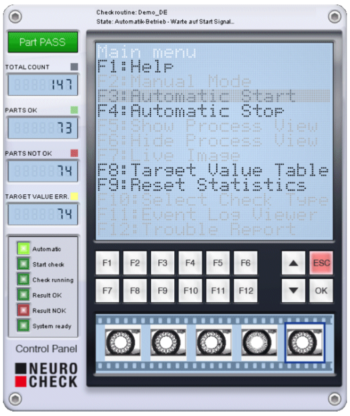
Depending on your configuration, only one or both windows can be visible on the automatic screen. The appearance of both windows can also vary greatly depending on the configuration.

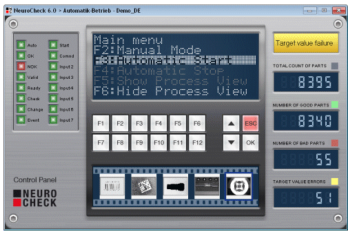
Using the Control Panel window, additional dialog windows can be opened with the help of the commands of the Control Panel Menu.

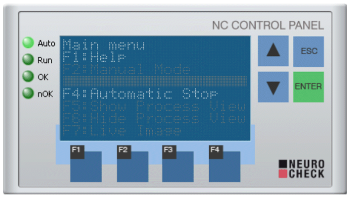
4.5.2 Control Panel

The Control Panel window is the main window for controlling the system in automatic mode. Here, the most important system information is visualized at a glance. Furthermore, you can interact manually with the system using a menu and buttons.

Depending on prior configuration, the layout of the window and the means for interaction can vary greatly. Here are some examples of how the window could look like:







Means for interaction

With a NeuroCheck system, the means for interaction with the operator can be designed in a very flexible way. This might be the reason why some of the following features are not available on your system:

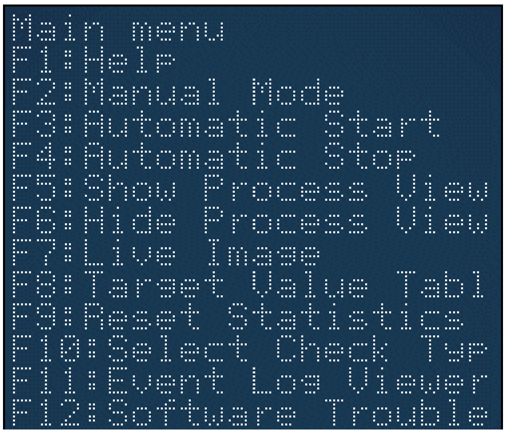
- Manual execution of commands via Control Panel Menu.
- Manual execution of commands via function keys and buttons.

The Control Panel Menu defines the actions or commands the user can execute in automatic mode and their assignment to the function keys.

4.5.3 Control Panel Menu Commands

The Control Panel Menu defines the actions or commands the user can execute in automatic mode and their assignment to the function keys.

Depending on prior configuration, the appearance of the Control Panel Menu can vary greatly. Here is an example of what the Control Panel Menu could look like:



Means for interaction

Here is a list of the most important commands: With a NeuroCheck system, the means for interaction with the operator can be designed in a very flexible way. This might be the reason why some of the following features are not available on your system or they have different names on your Control Panel Menu: Most of the following commands are only available in "Automatic stopped" mode, only a few in "Automatic started" mode:

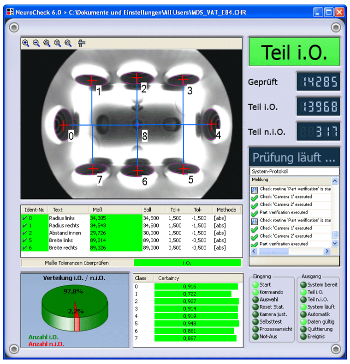
Name	Effect
Automatic Start	Switches from "Automatic stopped" mode to "Automatic started" mode.
Automatic Stop	Switches from "Automatic started" mode to "Automatic stopped" mode.
Automatic terminate	Terminates the check routine execution and switches to "Automatic stopped" mode.
Camera adjust	Opens the "Adjust Camera" dialog.
Check Routine save	Saves the current check routine to file. This saves all changes that have been made.
Check Routine select	Opens a dialog to select a different check routine.
Check Routine terminate	Terminates the execution of the check routine immediately.
Live image	Opens the "Live Image" dialog.

Manual Mode	Switches NeuroCheck to Manual Mode mode.
Process View show	Opens the Process View window.
Process View hide	Closes the Process View window.
Security profile select	Opens a dialog to select a different NeuroCheck security profile.
Self test	Opens a dialog to execute a check routine self-test.
Statistics reset	Resets the statistical values of the current check routine that are displayed in the Control Panel and Process View windows.
Target value table show	Opens the "Parameter / Target Value Input" dialog.
Target value table hide	Closes the "Parameter / Target Value Input" dialog.

4.5.4 Process View

The Process View window visualizes the process data of the current check routine (images, measured values etc.).

Depending on prior configuration, the appearance of the window can vary greatly. Here are some examples of how the window could look like:



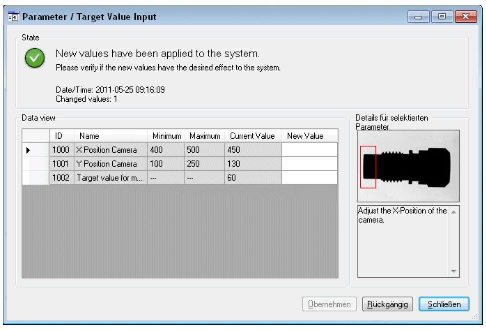
Means for interaction

With a NeuroCheck system, the means for interaction with the operator can be designed in a very flexible way. This might be the reason why some of the following features are not available on your system:

- Switch to Control Panel window using the button on the top left.
- 

Switch between various visualizations of Process View (so-called "Screen Layouts") using the list.
- Zoom and scroll within displayed images.

4.5.5 Parameter / Target Value Input



Open the **Parameter** / **Target Value Input Table** dialog by executing the Control Panel Menu Command called "Target value table". The dialog is used to change parameter and target values made accessible by publishing in the Parameter / Target Value Input Configuration dialog.

Depending on the settings made in the Parameter / Target Value Input Configuration dialog, the **Parameter** / **Target Value Input Table** dialog will display the editable parameters and target values for the individual items (check routine, individual check, check function) on individual tab pages or in a single table.

Changing a parameter or target value

- Select the tab page (if there are several) containing the value to be changed, either by clicking the tab or using the arrow buttons .
- Select the value to be changed by clicking it in the table.
- Enter the new value in the white column directly in the table. If display of parameter types has been activated in the Parameter / Target Value Input Configuration dialog, the area next to the input field will show the data type of the parameter. This determines, for example, whether you can enter only integral numbers or also floating point values.
- If the new value complies with an allowed range, it is marked yellow. If it breaks it, it is marked red.
- After entering one or more new values in the table, press the Apply button to apply them to the system. The system acknowledges the new values with a green flashing.

The dialog contains the following elements:

Element	Description
Input table	This list displays the published parameters and target values. Depending on the settings made in the Parameter / Target Value Input Configuration dialog the values for the individual check functions will be displayed on separate tab pages or together in a single table. This area displays the current value and is used for entering new values.
Type column	This area displays the data type of the currently selected parameter (visible only if parameter type display has been activated in the Parameter / Target Value Input Configuration dialog).
Info area	This area displays notes and instructions (only if comment display has been activated in the Parameter / Target Value Input Configuration dialog and if a comment exists for the currently selected parameter).
Apply	When this button is clicked, NeuroCheck will use the edited values starting with the next inspection run in automatic mode.
Undo	For a short period of time after pressing the Apply button, it is possible to undo the changes last applied. The time period can be configured in the Parameter / Target Value Input Configuration dialog.
Close	Closes the dialog box. Prior to that, edited values should be signaled to NeuroCheck using the "Apply" button, otherwise they will be lost.

4.5.6 Automatic Mode Start / Stop

NeuroCheck automatic mode can be switched between the states "started" and "stopped". This switching can only take place manually by using the Control Panel Menu.

System behavior for "Automatic Started"

- The systems reacts to digital input signals.
- If the signal "Start Check Routine" is active, the check routine is executed and the results reported.
- Most interactive commands of the Control Panel Menu are not available.
- The system sets the output signal "Automatic Mode enabled" to HIGH.

System behavior for "Automatic Stopped"

- The systems does not react to digital input signals.
- Most commands of the Control Panel Menu are available (exception: "Automatic stop" command).
- The output signal "Automatic Mode enabled" is set to LOW.

4.5.7 Terminate running Automatic Mode

If the system is in an infinite or waiting loop during the execution of a check routine (e.g. because it is waiting for an external signal that cannot arrive because of an external problem), it is possible to terminate the running Automatic Mode.

Termination of the Automatic Mode can be done in three ways:

- By the Control Panel Menu command "Automatic Terminate"
- By keyboard shortcut "CTRL+T"
- By the digital input signal "Terminate automatic mode"

Effect:

- A check routine being executed is immediately terminated.
- NeuroCheck also switches into "Automatic Stopped" state.

Please note that NeuroCheck may be in an undefined system state after executing this command, especially after multiple use. The reason for this is that software background threads will be terminated rigorously. In addition, the output signals may be in an undefined state after executing this command. So please use this command resp. signal only for emergency configurations.

Dissociation of "Check Routine Terminate"

In contrast to "Automatic Terminate" the command resp. signal "Check Routine terminate" only is the termination of the check routine. The system state remains at "Automatic Started".

5. Personalize Software

This section show various ways to personalize your NeuroCheck software and to adapt the system to the requirements of your application.

5.1 Arrange Windows and Toolbars

NeuroCheck manual mode offers you various ways to arrange the windows of the Development Interface according to your needs:

- Use the View menu to show or hide individual windows or parts of the user interface such as Info window, Quick Edit Table or Status bar.
- Some windows can be subdivided into several areas such as Check Routine Explorer or Result View.
- Some windows can be undocked and positioned anywhere as a floating window such as Info window or Check function selection dialog.
- The size of most dialog boxes can be changed by dragging the lower right corner with the mouse.
- Use the View menu to show or hide Toolbars.
- Use the mouse to change the position of the Toolbars in the upper window portion.

5.2 Software Settings

This section describes several options affecting behavior and appearance of the NeuroCheck software in Manual Mode and Automatic Mode in general.

You can access the dialog using the menu command **System** ▸ **Software Settings**.

5.2.1 General

5.2.1.1 Colors and Styles

Here, you can specify display colors and styles to be used for various items. Changed values are marked as bold.

The page contains the following elements:

Element	Description
Properties Table	List of object types for which colors and styles can be set. This dynamic list field contains the pertaining parameters (e.g. contour) in alphabetical order of the categories (e.g. ROI). The parameters of a category can be displayed or hidden using or . For a parameter, the value can be changed directly using the keyboard (please note formatting) or using the or icon on the right border of the cell. Exception: purely numerical fields. The information window can be found below the properties table and displays a short description for the currently selected parameter.
Reset to factory settings	Resets all alterations in all categories to default values via one mouse click.
Preview	Shows an example image with the currently selected color and styles options.
Apply	Saves all present alterations in this dialog. The dialog remains open for further modifications.

5.2.1.2 Start Behavior

Here, you can specify NeuroCheck's actions immediately after program start.

The page contains the following elements:

Element	Description
Intro (Default)	If this option is selected, NeuroCheck will display the intro screen at program start without loading a check routine.
Manual mode	If this option is selected, NeuroCheck will enter **Manual Mode** immediately after program start.
Reload last check routine at start-up	If this check box is selected, NeuroCheck loads the last active check routine in **Manual Mode**.
Automatic mode	If this option is selected, NeuroCheck will enter **Automatic Mode** immediately after program start.
Start immediately	If this check box is selected, NeuroCheck will switch to **Automatic Started** mode.
Load check routine from file	Displays the file name of the check routine which will be loaded for automatic execution immediately after program start.
Browse	Opens a file select box for selecting the check routine to be loaded.
Perform automatic self-test	If this check box is selected, NeuroCheck will perform the check routine self-test after loading the check routine in **Automatic Mode**. This option required the activated option **Load check routine from file**.

In license level **Runtime** NeuroCheck always starts in **Automatic Mode**.

5.2.2 Automatic Mode

5.2.2.1 Control Panel

5.2.2.1.1 Control Panel Design

Here you determine the Control Panel for controlling automatic mode. NeuroCheck supplies a number of predefined Control Panels that offer various selections of control elements.

You can also configure Control Panels individually if the predefined designs do not meet the requirements of your application.

The page contains the following elements:

Control element	Description
Designs	Select a predefined Control Panel.
Custom design	This selection necessitates that you have designed at least one user-defined Control Panel using the Control Panel Designer.
Preview	Shows a preview of the currently selected design.
Browse…	Only active when **Custom design** is selected. Opens a file selection dialog where you can integrate a user-defined Control Panel. (`*.cpx` file).
File name of custom design	Only active when **Custom design** is selected. Displays the complete path and file name of the user-defined Control Panel.

5.2.2.1.2 Control Panel Menu

Here you can determine the control menu of the Control Panel. NeuroCheck supplies a number of predefined default menus that offer various selections of functions.

You can also configure Control Panel menus individually if the predefined designs do not meet the requirements of your application.

The page contains the following elements:

Control element	Description
Menus	Select a Control Panel Menu.
Custom menu	This selection necessitates that you have designed at least one user-defined Control Panel using the Control Panel Menu Editor.
Preview	Shows a preview of the currently selected menu.

Browse…	Only active when **Custom menu** is selected. Opens a file selection dialog where you can integrate a user-defined Control Panel Menu. (`*.mnux` file).
File name of custom design	Only active when **Custom menu** is selected. Displays the complete path and file name of the user-defined Control Panel Menu.

5.2.3 Diagnosis

5.2.3.1 Logging

Here you can specify what debugging information NeuroCheck generates in Automatic Mode.

During automatic mode a large amount of data is written to the log file (e.g. 1 MB per second). Because of the permanent hard drive access, all other system executions are slowed down. Furthermore, dependent on your settings, the hard drive can reach its capacity limit fairly quickly.

The page contains the following elements:

Element	**Description**
Create log file	If activated, detailed log information is written to a log file in the NeuroCheck documents directory where NeuroCheck has been installed. An active logging is displayed with a symbol in the status bar.
High detail level	If activated, even more detailed information is written to the log file. Usually you will only activate this option if requested by the technical support or by the servicing staff.
View log file	This button opens a file select dialog, which enables you to view a log file.
Delete log files	Clicking this button opens a file select dialog that enables you to mark certain log files for deletion. Only marked files will be deleted!
File name	**Standard**: In this mode, NeuroCheck always generates the log file as fix file name `NcLOG.XML`. If the file reaches the maximum file size specified, the contents of the file are cleared and the file is written again. **Date and Time**: In this mode, a new log file is generated at each start of the NeuroCheck software. The file name is created by using the current date and time information. If the file reaches the maximum file size specified, a new file is generated by using the current date and time information and so on. In theory, an infinite number of files will be generated. **Ring buffer**: In this mode, a new log file with a unique ID (e.g. `NcLog_001.xml`) is generated at

	each start of the NeuroCheck software. If a file reaches the maximum file size specified, NeuroCheck switches to the next ID file. If the number of log files specified in **Files** is reached, the log file with the index 001 will be overwritten etc.
Properties table with information window in "File options"	Here you can specify the properties of the log file. **File size (in kB):** Limits the size of the log file. **Format:** Specifies the output format of the log file (`*.txt` or `*.xml`). If you selected ***.xml** as output format, you can change additional settings in the **XML visualization** category. These settings affect only the amount of data displayed of the log in a Web browser. It has no effect on writing the data to a log file. (All data is written to the log file.)
Info field	Corresponding to the selected row in properties table, this field displays a brief description.

5.2.4 Plug-Ins

5.2.4.1 Plug-In

Here, you can managing plug-in DLLs which add user-defined functionality to NeuroCheck.

> Please note that some settings can only be changed if the current check routine does not use plug-in check functions.

The page contains the following elements:

Element	Description
List of plug-in-DLLs	The list shows the currently loaded plug-in DLLs with their version numbers. DLLs where the check box in front of the name is unchecked will not appear in the **Select Check Function** dialog in check function group **Plug-In**. Their menu items will also not appear in the **Tools** menu.
Add	Opens a file selection dialog for loading a new plug-in DLL.
Remove	Select in list via left mouse button the DLL, which you would remove from system. Clicking on button **Remove** deletes your selection. You cannot remove more than one DLL from list simultaneously.
	Change the loading sequence of plug-in DLLs, in case one DLL is dependent on another.
Details...	Displays the information dialog of the plug-in DLL selected in the list.

5.3 Security Profiles

This section describes how to individually restrict access to the NeuroCheck system.

5.3.1 Security Profiles: Introduction

In addition to the protection of individual check routines using the Password Protection dialog, NeuroCheck supports establishing a user-defined number of security profiles. A system administrator can individually restrict access to certain functionality for each profile. This system allows OEMs or production supervisors to protect their installed applications against any unauthorized use. Furthermore, all activities can be logged by user name.

Switching between security profiles

There are two ways to switch between two different security profiles:

- By keyboard and entering a password in the Switch Security profile dialog.
- By external digital signals that can be configured in the remote control input signals dialog.

Realization of the security profile system

For maximum security the system of access profiles is managed through a combination of dongle programming and a password file. The fact itself that a NeuroCheck installation is password protected is encoded in the dongle, the current passwords and the default access level are stored in a configuration file in the NeuroCheck project directory. This has the following consequences:

- The program will start only if the correct combination of dongle and password file is present.
- Removing the password file prevents the program from running at all, because the dongle encoding is preserved.
- Removing the dongle converts the software into a Demo Version unusable for automated visual inspection systems.

The only way to remove the protection without the system administrator password is exchanging the dongle.

> For establishing a security level system as well as for removing it, a Premium or Professional license level dongle is required (see section "NeuroCheck License Levels" for details). Also note that password files cannot be exchanged between NeuroCheck installations, because the password file depends on the dongle used to create it.

- **Establishing a security level system**

  The security level protection is established by choosing **Security Profiles** from the System menu, entering some security profiles in the Security Profiles dialog and leaving the dialog with "OK". A step by step introduction is given in section "How to establish security levels".

  As soon as the Security Profiles dialog is left with "OK", the dongle is reprogrammed by NeuroCheck to encode the fact that this installation is now password protected. Simultaneously, in the NeuroCheck directory, corresponding entries are generated in the file "NC60CFG.SP.CFGX", recording the passwords and the current security profile. Note that you will have to be working within a security profile with system administrator rights to add, edit or remove security profiles and passwords.

- **Start-up procedure of a protected installation**

  Upon program start-up NeuroCheck queries the dongle for the protection flag. If the protection flag is set it searches for the password file. When the password file is found, the defined security profiles and their passwords are read and the most recently used security profile is activated. This ensures that noone can enter a security level giving him more rights than are due to him without having the correct password.

- **User permissions**

  You can add and edit security profiles and their permissions in the Security Profiles dialog.

- **Removing system password protection**

  Leaving the Security Profiles dialog while no security profile is activated in the list view removes password protection. The dongle is reprogrammed. Only a system administrator has the right to do this.

- **Emergency Recovery**

  When the password file of a protected NeuroCheck installation is destroyed the system will not start. In this case there are two ways of recovery, both explained in detail in section "Security level recovery":

  - Reprogramming the system using a non-protected Full Version dongle.
  - Contacting your software vendor for a context file to reprogram the dongle.

5.3.2 Security Profiles Configuration Dialog

The Security Profiles dialog is displayed upon choosing **Security Profiles** from the System menu. It allows defining and editing security profiles. This function is available only under the following conditions:

- You are running a NeuroCheck Fully Licensed Version (Premium or Professional Edition). See section "Versions of NeuroCheck" for details.
- The current security profile has system administration rights (or there is no security profile system active).

Set up administrator authorizations (permission to edit user profiles) in dialog "Basic Settings" on page "Others".

A step by step introduction is given in section "How to establish security profiles".

The dialog has the following elements:

Element	Description
Profile list	Lists the currently defined security profiles. Only profiles with an activated check mark in front of the profile name are accessible in the Extra toolbar. Leaving the dialog with "OK" while no profile is checked deactivates the security level system.
Add...	Opens the Security profile settings dialog for adding a new security profile.
Edit...	Opens the Security profile settings dialog for editing the security profile currently selected in the profile list.
Duplicate	Generates a new security profile with identical properties and permissions like the security profile currently selected in the profile list. The identification number will be different though.
Delete	Deletes the profile currently selected in the profile list.
Options…	Opens the Security Profiles Options dialog. There you can configure global settings for all security profiles and for the security profile system in general.

5.4 Managing Projects

This section describes possibilities and tools for managing several projects in NeuroCheck.

5.4.1 Introduction to NeuroCheck Projects

A NeuroCheck project consists of all files and directories necessary to operate an inspection system. These are mainly:

- Inspection routines
- System configuration
- Sample data
- Input data
- Output data
- Log files and event log files

Memory location

When installing NeuroCheck on a computer, configuring the system and check routine in manual mode, and executing them in automatic mode, this takes place in a project named **Default** without you being aware of this. This is only important if you want to create a backup of the project data or if you want to run several projects parallel on one computer.

5.4.2 Import Configuration Dialog

In some cases, it can be advisable to copy the system configuration of another Project e.g.

- to transfer a project having been prepared in the office to the visual inspection line;
- to create a similar project;
- to analyze and optimize a client's existing system.

NeuroCheck enables you to import the entire system configuration or selected parts thereof from an existing Project. For this select the **Import configuration** dialog box from the **System** menu. Since the NeuroCheck system configuration is divided into two configuration files (one for data processing and one for the user interface), two existing configuration files must be selected for import.

> By importing another configuration, the current system configuration is deleted and replaced with the selected configuration. The old system configuration is not recoverable.

The dialog has the following elements:

Element	Description
Data processing file	Select a configuration file for data processing from (`file name: nc60cfg.DP.cfgx`)
User interface file	Select a configuration file for the user interface (`file name: nc60cfg.UI.cfgx`)
Data to import	Select the parts of the system configuration you want to import from the above selected configuration files into the current configuration. The unselected parts remain unchanged during import. If only parts of the configuration are imported, the configuration may not be valid or operational afterwards, e.g. if signals are configured in remote control without any communication devices being configured.
Apply and Restart	If you press this button, the selected parts of the system configuration will be replaced. For this you need to exit and restart the software.

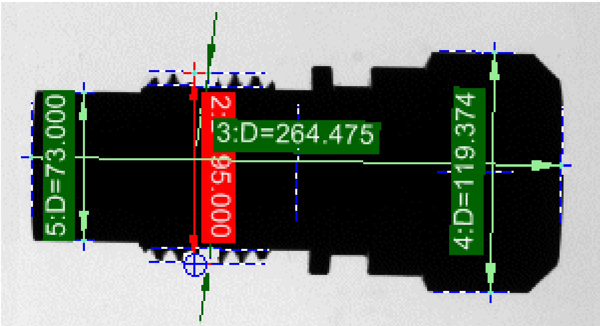
6. Special Applications

This section covers advanced and special topics.

6.1 Gauging

6.1.1 Introduction to Gauging

The check functions in group "Gauging" are used for creating and processing complex geometrical measurements contained in measurement lists.



Gauging rules

A gauging rule determines the kind of measurement computed by function **Gauge ROIs** for one or more objects. The applicability of a gauging rule depends on the number of selected objects and on the geometrical description. chosen for the objects. Measuring the waviness e.g. requires the presence of a contour and a line as model geometry, measuring distances requires two objects etc. Object selection and assignment of gauging rules is done in function **Gauge ROIs**.

Geometrical properties

NeuroCheck uses geometrical descriptions of objects for gauging. These can be actually existing properties of the objects, e.g. the coordinates of its center of gravity, but also properties of model geometries, computed from one or several objects.

Subpixel precision

Special interpolation algorithms allow the partial reconstruction of information lost due to the quantization inherent in digital images. Thus it is possible to measure with a resolution better than one image pixel.

6.1.2 Geometrical Descriptions

Every object in NeuroCheck can be described by various geometrical properties. All objects own the geometrical descriptions **Point** and **Contour**, representing its center of gravity and its silhouette. In addition, model geometries can be computed, approximating a piece of a contour by some ideal geometrical structure, e.g. a circle or a straight line.

Every object created by NeuroCheck (e.g. using check function **Create ROIs by thresholding**), owns the following geometrical descriptions:

- **Point**:
  Coordinates of the object's center of gravity or the center of a model geometry, if one has been computed for the object;
- **Contour**:
  The actual silhouette of the object (note that an object created by function **Template Matching** uses the borders of the template as its contour, as there need not be a visually connected silhouette inside the template).

Using check function **Compute model geometries** the following model geometries can be generated:

- **Line**:
  Optimal approximation of an object contour or a line through the centers of a group of objects by a straight line.
- **Circle**:
  Optimal approximation of an object contour or the configuration of an object group by a circular arc.

Model geometries can either be computed for single objects or for whole groups of objects. Thus it is possible to calculate a circle from the centers of gravity of three objects (exactly) or more objects (approximately), or a straight line from two objects (exactly) or more (approximately). Computation of model geometries is done by function **Compute model geometries**.

Function **Gauge ROIs** allows you to decide for every object individually, which geometrical description to use for gauging. The setting of the geometrical description determines the measurements which can be computed. The available measurements and their relation to geometrical descriptions are described in the following sections.

6.1.3 Gauging rules

6.1.3.1 Gauging Rules: Summary

This section summarizes all available gauging rules in NeuroCheck together with their requirements on the object configuration to be measured. Note that the exact meaning of a gauging rule depends on the selection of geometries to be related to each other.

Name	Number of Objects	Geometries
X coordinate Y coordinate	1	Point
Enclosed area	1	Contour
Radius, average Radius, minimal Radius, maximal	1	Contour or Circle
Diameter, minimal Diameter, maximal	1	Contour
Extension, maximal	1	Contour
Line angle Waviness	1	Line
Diameter Roundness	1	Circle
Distance Distance directed ± Distance directed	2	Point to Point
Distance, minimal Distance, maximal	2	Point to Contour Point to Circle Contour to Contour Contour to Circle Line to Line Line to Circle

Distance directed, minimal Distance directed, maximal	2	Point to Contour Contour to Contour
Distance opposite, minimal Distance opposite, maximal	2	Contour to Contour
Distance absolute Distance, signed	2	Point to Line
Distance, perpendicular minimal Distance, perpendicular maximal Parallelism	2	Contour to Line
Intersection, X coordinate Intersection, Y coordinate Intersection, angle Distance, centers	2	Line to Line
Concentricity	2	Circle to Circle
Angle at Point 0	3	Three points
Distance, line intersection to contour, minimal Distance, line intersection to contour, maximal	3	Two lines to contour
Line length	3	Three lines

6.2 Classification

This section covers the classification basics and describes how to use this concept in NeuroCheck.

6.2.1 Classification

"Classification" denotes the assignment of one of several predefined classes to an input signal.

Examples:

Application	**Input signal**	**Class description**
Distinguishing objects (e.g. screws)	• Object image • Object measurements	• Object name
Distinguishing surface defects	• Image of the defective area • Dimensions of the defective area • Brightness of the defective area • …	• Name of the defect (scratch, hole…) • Assessment of the defect as acceptable or unacceptable
Character recognition	• Image of the character • Fourier descriptors of the characters • Shape properties of the characters (like dimensions, curvatures...) • …	• Name of the character • Number of the character in a predefined set of characters
Acoustic inspection of an electric motor	• Digitized signal from microphone • Spectral properties of the signal • …	• Assessment of the motor as gooc or faulty

It is apparent that even in the technical area alone very different types of classification problems exist. The common element of all these applications is a significant reduction in the amount of data. The input signal usually contains a large amount of data (an image of a character should consist of at least 200 pixels, better 400, to be safely recognizable; the spectral properties of acoustic signals are frequently described by at least 256 spectral lines). Every single component of the input signal is called a feature. Every combination of feature values describes a specific object. The number of classes, on the other hand, is significantly smaller.

This leads to the following definition:

Classification is the assignment of a designation (class) from a set of possible classes to an object characterized by a set of feature values (which is usually significantly larger than the set of classes).

The actual classification can be performed in many ways; some of them are briefly introduced in section "Types of classifiers".

6.2.2 How to use Classifiers in NeuroCheck

6.2.2.1 Using Classifiers in NeuroCheck

The following sections give a brief introduction into using classifiers in NeuroCheck. In order to achieve a largely self-contained explanation, some remarks from the previous sections will be repeated here.

Classification tries to model aspects of human reasoning, therefore it is a complex subject. NeuroCheck makes applying a classifier to a problem as easy as possible, performing many of the tasks necessary for creating and using classifiers automatically, but nevertheless a specific procedure has to be observed and some thought has to be invested in how to make the best use of this powerful technology.

When to use a classifier

Not every problem needs a classifier. If the distinction between objects of different classes can be made by simply comparing some measurements to certain thresholds, a screening process will be sufficient – as it is realized by function Screen ROIs. As soon as there are more complex, possibly non-linear relationships between measurements, one might think of using a classifier. The same holds, when not only simple features, but the overall appearance of the object has to be taken into consideration as is the case in character recognition.

Other ways of generating class information

Standalone classifiers (neural networks in NeuroCheck) are not the only way to generate class information. For example, after an object has been found using Template Matching NeuroCheck has the information, which template was most similar to the object and thus is able to attach a class to the object, namely the class of the template. Note, however, that the correlation algorithm used by Template Matching is by nature a linear classifier and therefore not as powerful and robust as a neural network.

The following check functions are currently able to generate class information:

- Classify ROIs
- Template Matching

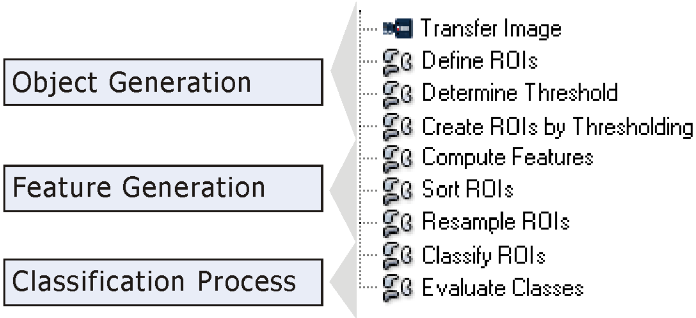
6.2.2.2 Features for Classification

In order to fulfill his recognition task, a classifier needs features describing the objects to be recognized. The kind of features appropriate for a specific recognition task depends on the application. In character recognition a normalized image of the character will frequently be used, as is it is produced by function Resample ROIs. If, on the other hand, the task is some kind of shape distinction, measurement values, as computed by function Compute Features, will be more appropriate. A complete list of functions, which add features to regions of interest, can be found in section "Object features".

Check configuration

The first step in using a classifier is, therefore, to generate its input data by one of the functions mentioned above, depending on the type of recognition task. Then you will have to append function Classify ROIs to the check.

Such a check will look something like the following image:



6.3 Data Register Manager

6.3.1 Data Register Manager

The Data Register Manager dialog serves to configure the data register cells that you can use for system's data input and data output.

Using the dialog, you can create, edit and view Registers and their connections. The entire data exchange from the check routine to peripheral equipment and vice versa takes place via the registers. Each individual value is moved into a register and then picked up from the other side.

Thus you can realize dynamic value setting using the Input Registers and result output using the Output Registers. Because of the buffering in the registers, it is possible to use the values in the registers for calculations.

All registers are defined with their properties and connections to peripheral equipment In the Data Communication Center. On the peripheral side, each register is connected to a data pin of a Data Format Converter. The connections to the check routine side are configured using the data input and data output dialogs.

6.3.2 Register

Function Description

The registers are data sheets or "interim storage" for keeping data and manipulating it freely. Registers are created and edited in the **Data Register Manager**.

Register Categories

The registers in the Data Register Manager are divided into three different register categories:

- The **Input registers** are used for dynamic value setting.
- The **Output registers** are used for data output.
- The **Free registers** are used for any kind of data manipulation using the check functions **Modify Data in Register** and **Verify Data in Register**.

Data types

Each register cell has a fixed data type. This determines what kind of data can be stored in the register cell:

- String
- Integer (integral number)
- Float (floating point number)
- Boolean

Operation in Automatic Mode

- **Input Registers**:
  are used for dynamic value setting. At the start of the entire check routine execution, the input registers retrieve their new data from the connected input data pin. Range checking for the values can be configured. At the start of the execution of a check routine object (e.g. a check function), the object retrieves its accessible data from the connected input register.
- **Output Registers**:
  are used for data output. At the end of the execution of a check routine object (e.g. a check), the object writes its published data into the connected output registers. At the end of the execution of the entire check routine, the data is sent from the output registers to the connected output data pins.
- **Free Registers**:
  There is no special treatment in automatic mode.

Operation in Manual Mode

Usually the Data Registers are not used in manual mode. To activate/deactivate, select **Use data registers in Manual Mode** from the **Tools** menu.

6.4 Data Format Converter Manager

The Data Format Converter Manager manages all active Data format Converters in the system.

For all data and signals going from the system to the periphery or vice versa, a Data Format Converter must be configured for transmission. A Data Format Converter corresponds to a communication channel. The transfer direction can be input, output or bidirectional.

Data from/to the check routines are processed in the data registers. Further use of the process signals, however, is configured in the Remote Control Dialog.

Each Data Format Converter has properties that can be managed using the Data Format Converter Manager. The main properties of each Data Format Converter are the link to a hardware device (e.g. digital I/O, field bus, serial communication, Ethernet communication) or a file and a list of input and output pins. Each pin is a fundamental data cell provided by the Data format Converter for use in the program. The Data Format Converter converts the format of the external periphery into the internally required format.

7. Appendix

7.1 License Levels

This section describes the different licensing levels of the NeuroCheck software, and how to manage them.

7.1.1 NeuroCheck License Levels

The license level is encoded in the hardware key (dongle). The hardware key can be re-programmed to change the license level.

The following editions (license levels) of NeuroCheck are available:

Premium Edition

Complete functionality for configuring check routines in manual mode and full hardware access, i.e. several cameras, digital and serial communication. In addition, the Premium Edition offers extensibility by plug-in functions and custom communication interface drivers. The Premium Edition can be protected against unauthorized use.

Professional Edition

Identical to Premium Edition, apart from the restriction that no check routines can be configured using plug-in check functions. The Professional Edition can load and execute check routines with plug-in functions, but it does not allow adding plug-in check functions to a check routine. In addition, the Plug-in software extension wizard is not available in the Professional Edition.

Runtime Edition

This license level is intended for the duplication of completely configured inspection applications. Like the Premium or Professional Edition, it has complete image processing, hardware access and extension functionality, but restricts operation to automatic mode. No configuration of check routines or system parameters is available in a Runtime Edition. Reconfiguring a check routine in the Runtime Edition requires exchange of the Runtime hardware key against a Professional or Premium licensed key and restarting the software. Possession of at least one Professional or Premium license is required for acquiring a Runtime license.

Demo Version

NeuroCheck automatically runs as a demo version when the hardware key is removed. The demo version allows no access to special hardware devices. There is no communication, remote-control or extension functionality nor is it possible to capture images from a camera. This non-licensed version is intended for evaluation purposes only. Regular use of a demo version in an image processing laboratory or in an automated visual inspection system is illegal. Please see section Demo Version for details.

Software feature availability in the editions

Software-Feature	Runtime Edition	Professional Edition	Premium Edition
Check in automatic mode	X	X	X
Use any number of cameras	X	X	X
Integrate system into process environment	X	X	X
Adjust target values and parameters during operation	X	X	X
Build check routines interactively		X	X
Use interactive image analysis tool		X	X
Protect system with user profiles		X	X
Create user-defined functions (plug-in)			X
Use a wizard to create plug-in functions			X

7.1.2 Demo Version

Without a hardware key, NeuroCheck runs as a Demo version. You can use the full built-in image processing functionality of NeuroCheck to evaluate the software with the Demo version. The only restrictions apply to the use of the software for building and running automatic visual inspection systems.

Restrictions of the Demo version

- No image capturing from camera.
- No digital or serial communication.
- No result output to file.
- No extensibility by plug-in functions or custom communication protocols.
- No saving of system configuration settings.

The evaluation license does not include printed documentation. Please understand that we can not offer our full service free-of-charge technical support telephone assistance for users of the Demo version.

Legal notes

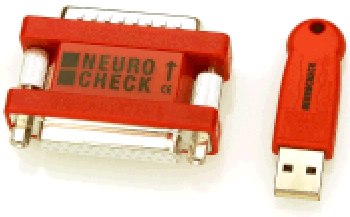
Title, ownership rights, and intellectual property rights in the software shall remain in NeuroCheck GmbH. Software and documentation are protected by the copyright laws and treaties.

The Demo version of NeuroCheck is intended for evaluation purposes and private use only. To this end it enables you to test the full extent of NeuroCheck's image processing functionality. Any other use of this program version is illegal, in particular application in automated visual inspection systems and regular use in laboratories or research facilities. Making use of results obtained by using a demo version in technical or scientific publications other than software reviews is illegal and violates the copyrights of NeuroCheck GmbH.

Please refer to the Copyright and Support Services sections of the NeuroCheck online help system for information on ordering unrestricted versions.

7.1.3 The Hardware Key (dongle)

Every NeuroCheck version comes with a hardware key (a so called "dongle"). Licensing is done before delivery of the software package by programming the hardware key. The hardware key constitutes the actual value of the purchased NeuroCheck license. Two different types of hardware keys are available: one for the parallel interface (LPT) and one for the USB interface of a PC.



The license level as well as the NeuroCheck version (e.g. 6.0, 6.1) are encoded in the hardware key.

License level or version upgrades can easily be carried out remotely via email without having to physically exchange the hardware key. Please refer to section Dongle Upgrade.

> Without a hardware key, NeuroCheck runs as a Demo version.

7.1.4 Dongle Upgrade

License level or version upgrades can easily be carried out remotely via email without having to physically exchange the hardware key. The necessary steps for re-programming the hardware key are initiated conveniently from the Device Manager.

Purpose of the Dongle Upgrade procedure

The hardware key upgrade process has to be performed:

- For an Upgrade of your NeuroCheck version (i.e. from version 6.0 to 6.1).
- For an Upgrade of your license level (i.e. from Professional Edition to Premium Edition).
- In order to support non-standard features for special applications.

> Please do not forget to send or fax a written purchase order for the license update to your NeuroCheck dealer.

7.2 Event Log

7.2.1 Event Log: Introduction

The event log helps to analyze problems with the system occurring, for example, because of failures or operating errors during production.

What is the Event Log?

NeuroCheck stores all important system, security and user events in an event log with the name `EventLog.dat` in the NeuroCheck System Output Directory. The event log is persistently stored, thus preserving the data after closing the NeuroCheck software.

What are the benefits of the Event Log?

The `EventLog file` created by the software is displayed in the Event log viewer dialog. You can use this information for monitoring, maintaining and optimizing the system.

The Event log viewer enables quick offline error analyses using the import and export functionality. Export the `EventLog file` to mail it, for example, to the technical support or the responsible project engineer. This enables fast troubleshooting since the person responsible does not necessarily have to be on site for error analysis.

What are the differences between Event Log and Log File?

The event log must not be confused with the Log file; the creation of the log file must be explicitly activated. The event log is always created automatically by the software and, unlike the log file, does not record all data but only important system events.

For further configuration settings for both logs see the main menu **System** ▸ **Software-Settings** ▸ **Diagnosis**.

7.2.2 Events

Events are occurrences in the system, application or check routine area. There are four types of events:

- **Error:** severe errors impairing system performance
- **Warning:** errors or potential problems. Maintenance of the system is recommended.
- **Audit Trail:** user intervention in the system or check routine configuration
- **Information:** normal operation events, e.g. the check routine result

This information is stored for each event:

- Event type:
  (Error, warning, audit trail information or information)
- Category:
  Origin of event information (e.g. system, user interface, check routine etc.)
- Event text:
  Text describing the occurred event.
- Identifier:
  It is language-independent and used for the traceability of NeuroCheck event information (e.g. when using the NeuroCheck interface with another language)
- Event details:
  Specifications or details regarding the event information (not available for all events).
- User:
  Name of user logged into the system when the event occurred (if Security Profiles are activated)
- Date and Time:
  Point in time when the event occurred.
- Module description:
  Internal identification number of the software module registering the event information.

7.2.3 Event Log Viewer Dialog

Use this dialog to get an overview of the Events that have occurred or to search for events using filters.

Dialog structure

The event display is usually subdivided into three areas:

- The left side of the dialog contains the four information types in a tree structure.
- The right side contains all events in table form at the top (depending on the selection in the tree).
- On the bottom of the right side either details concerning the selected event are displayed or graphical statistics relating to the current selection.

Using event log viewer dialog

- **Event type view:**
  Using the tree view on the left, all events or just the events of one specific type can be selected:
  - All events
  - **Error:** severe errors impairing system performance
  - **Warning:** errors or potential problems. Maintenance of the system is recommended.
  - **Audit Trail:** user intervention in the system or check routine configuration
  - **Information:** normal operation events, e.g. the check routine result

- **Sorting within the table:**
  Click on a column header of the table in the upper right to sort the events in ascending/descending order (e.g. according to "date and time").
- **Event details:**
  Details concerning the currently selected event from the table are displayed in the event display in the lower right corner. You can also configure columns for display in the table using the dialog menu **View ▸ Show/Sort columns…**.
- **Show/Hide Diagram** or **Properties**:
  On the lower right of the dialog you can either display the properties of the current event or diagrams.
  - Diagram:
    Depending on the selection in the tree structure, the diagram contains the five most frequent events of a specific event type or, if "All Events" is selected, the number of occurrences of events for each event type.
  - Properties
    Only the properties of one selected event can be displayed. From Event Properties, you can copy the event details to the clipboard using the **Copy** button for further use in an error report, e-mail etc.
- **Filter:**
  Define various display filters to narrow down the contents of the event table. Click on the filter icon or **View ▸ Filter…** in the dialog menu.

  > The filter can only be used for all events. It is not possible to filter events of a specific information type.

- **Export and import of event logs:**
  When opening the dialog, the Event Log of the current software system is displayed. Use the **File** dialog menu to export the current event log as a file (*.dat) or to import a *.dat file from another system.

Other toolbar icons

Element	Description
Back	Returns to the previously selected event type.
Next	Switches to the subsequently selected event type.
Select parent directory	Switches to the main directory where all event types (irrespective of a selected filter) are displayed.
Directory structure view/hide	Displays or hides the left window frame (containing the event explorer with the tree structure).

7.3 Software Trouble Report

7.3.1 Software Trouble Report

To facilitate an effective problem analysis by the manufacturer's technical support, it is often necessary for you to make a trouble report and send it to support. You can create this trouble report directly within the software using a wizard. If the computer is connected to the Internet, you can mail the trouble report directly from the software.

If you previously entered all company and mail server data in the **System** ▸**Software Settings** dialog in section **Diagnostics** ▸**Contact information**, this information can be used directly when calling up the trouble shooting wizard. Here you can also change the recipient's e-mail address if you don't want your customers to contact support@neurocheck.com directly.

Troubleshooting report content

The troubleshooting report contains information on the company and contact person, a problem description and information on the system environment. In addition, files can be attached to the report that are necessary for analysis by the technical support. Before you send the data to the technical support, you can view the entire troubleshooting report.

The troubleshooting report is created in four steps with the wizard asking for the following information:

- Personal Information
- Problem Description
- Attachments
- Send Report

7.4 Glossary

ADC

Analog to digital converter, converts analog voltages to numerical values, e.g. the voltages generated be light falling on a CCD-sensor to gray levels used in digital image processing.

AOI

Area **o**f **I**nterest. Outlining rectangle of the region of interest. Please don't confuse with the abbreviation of the definition "automatical optical inspection".

Artifact

Objects that do not exist in the captured real-world scene but were created because of certain properties of the image acquisition or processing (e.g. artificial contours resulting from coarse quantization).

Automatic Mode

In automatic mode NeuroCheck executes a check routine upon receiving a start signal (key press, timer, digital I/O, serial interface, or other communication channels) and reports the check result "OK"/"not OK". It is also possible to switch between different check routines by external signals.

Automatic Screen

In automatic mode the screen layout of NeuroCheck can be individually configured for every check routine. A check routine can manage several different screen layouts for different purposes.

Bar Code Type 128

The code 128 consists of bars of multiple widths, with eleven modules for each symbol. Both stripes and blanks are of variable width.

Bar Code Type 2/5 Interleaved

The code 2/5 interleaved can contain only an even number of digits. Consisting of bars and interspaces in only two different widths it is easy to identify even under poor conditions.

Bar Code Type 39

The code 39 is one of the few bar codes able to represent letters in addition to digits. Because of this it requires a relatively large space for each character.

Bar Code Type EAN13

The EAN 13 code is used for most products throughout Europe, e.g. books, magazines, food,... It starts and ends with two pairs of bars slightly longer than the rest. Another such pair divides the code in half. The code contains 13 digits. It consists of bars and spaces in four different widths and is therefore difficult to identify under poor conditions.

Bar Code Type EAN 8

The EAN 8 code is similar to the EAN 13, but contains only 8 digits. It is frequently used on cosmetic products in Europe.

Bar Code Type PZN

The PZN code consists of a minus sign, six digits and a check digit. It is used on medical packages in Germany.

Bar Code Type UPC-A

The Universal Product Code is a subset of the EAN-Code. It is used for most products in the U.S., e.g. books. magazines, food... It contains 12 digits.

Bar Code Type UPC-E

This Universal Product Code is a shorter version of the UPC-A-Code. It is used on food and cosmetics in the U.S. It contains 8 digits.

Binary Image

A binary or bi-level image contains only black and white pixels. It can be computed from a gray level image by thresholding. Each gray value above the threshold is set to the gray value 255, representing "White", all others to 0 for "Black".

Bitmap File

Bitmap files (extension "BMP") are the standard file format for pixel images in Windows. They can be inserted or pasted over the clipboard in most Windows applications.

CCD

Charge-coupled device, integrated semiconductor device that operates by passing along electric charges from one stage to the next. The charges can be generated by incoming light. Matrixes of CCD elements are frequently employed as image sensors in video cameras.

Calibration

Determination of a conversion factor between pixels in the digitally recorded image and real world measurements in the form of metrical units.

Camera Setup File

Configuration settings for a digital camera or the camera input channel of a frame grabber can be exported to a camera setup file. Thus it is possible to easily transfer camera configuration settings from one input to another or even from one computer to another. Importing and exporting camera setup files is controlled from the Device Manager dialog.

Certainty

The certainty in a classification operation is defined as the ratio of the difference between the highest and the second highest class output to the sum of both. This evaluation of the certainty is based on the observation that a misclassification by a well-trained classifier will practically always create two outputs of similar magnitude, never a very high one in a wrong class.

Example: a good classification will have a single high output, e.g. 0.95 (outputs of exactly 1.0 are extremely rare and not necessarily preferable), whereas the outputs of all other classes will be very low, e.g. less than 0.05. In this case, the certainty would compute to (0.95 - 0.05)/(0.95 + 0.05) = 0.90. A bad classification will frequently exhibit two outputs close to 0.5, e.g. one at 0.6, one at 0.4, giving a certainty of (0.6 - 0.4)/0.6 +0.4) = 0.2.

Note: for a template matching classifier, the certainty simply equals the correlation coefficient between the region and the best matching template.

Check

 This symbol depicts a check in the structure view in manual mode.

An individual check (short: check) solves a specific part of the complete inspection task, e.g. reading an inscription, measuring an object or checking the presence of a part.

A check consists of a sequence of check functions. Each check function performs one system, image processing or classification function. Checks can be individually imported from complete check routines, allowing you to re-use proven solutions for known inspection tasks without having to reinvent them every time.

Check Function

A Check Function is an image processing function, a classification function or a system function that can be used within a check to fulfill a specific processing task, e.g. generating regions of interest by thresholding.

The available check functions are divided into different check function categories.

Depending on the category a check function belongs to, it will be depicted by its corresponding symbol.

Check Function Group

The check functions in NeuroCheck are divided into groups according to their processing tasks. An overview of these groups is given in section "Check function groups".

Check Routine

 This symbol depicts the check routine in the structure view in manual mode.

A check routine comprises the complete solution of an inspection task. It is the central element of every image processing application realized with NeuroCheck. Its role can be compared to that of a text document in a word processor or a source code in a development environment.

Usually a check routine consists of several individual checks, that are responsible for self-contained parts of the check problem, e.g. reading an inscription, measure an object or check the presence of a component.

Check Routine ID

Number attached to a check routine on the Communication property page of the Check Routine Properties dialog box. These numbers are used to switch between check routines in automatic mode upon an external signal.

Check Routine Listing

A listing of the individual checks of a NeuroCheck check routine with their check functions can be printed with the function **Print** in the Check Routine menu.

Check Routine Window

The Check Routine Window displays in its left pane a hierarchical tree view of the individual checks of a NeuroCheck check routine with their check functions. The right pane displays the result of the check function most recently executed. Many check functions offer several different modes of displaying the result in a context menu. The displayed result can be transferred to the clipboard via the Copy function.

Classifier

A classifier assigns a class name to objects represented by features (i.e. measured quantities). An example is the recognition of a letter from a certain configuration of image pixels. The features in this case are the brightness values of the image pixels, the letter is the class name.

Color Image

A color image usually contains color information represented by the three color channels Red, Green and Blue, which each have a range of [0..255]. Those check functions that are not able to process color images, internally use a gray value channel that is derived from the color image.

Color value

A color consists of three color values. For RGB color images the three color values are corresponding to the channels of the color space: red, green and blue. Each color channel has a valid range [0..255].

Contrast

Measure for the uniformity of the gray level distribution within an image. An image with a large number of very bright and very dark areas has a high contrast, an image containing a broad range of gray levels is of medium contrast. If the image consists of a small number of distinct gray levels from a narrow range of the complete gray level spectrum, its contrast is low.

Convolution

In image processing the multiplication of two matrices, one usually a section of the image, the other a convolution kernel that is shifted across the image in steps of one pixel. This operation is equivalent to a linear filtering of the image.

Data Object

Check functions within a NeuroCheck check routine store result data as data objects of different data types (like gray level images regions of interest, measurement lists) in the data pool, where subsequent check functions can access them.

Data Packet

In NeuroCheck every value transmitted by a check function via the serial interface forms a data packet, including control information like check and function index. The NeuroCheck standard protocol wraps a data packet in the control characters "Start of Text" and "End of Text". Compare to records generated for output to file.

Data Pool

The data pool collects all result data objects created by the check functions of one particular check within a NeuroCheck check routine.

Data tray

Global storage area for images, list of regions of interest, measurement lists and histograms. The Data Tray is managed using special Check Functions. The data tray can be used, for example, to use objects from the data pool in a different check or another check routine.

> The data tray objects are only stored in RAM. At the end of the program, all data tray data is lost!

Decision Check Function

A decision check function compares results of an image analysis with nominal values (target values), e.g. a bar code or identified characters with a prescribed string. It returns OK/not OK as its result. In the case of "not OK", the execution of the check stops.

Digital I/O

Digital I/O is used mainly for communication with the PLC and uses 24V or TTL level signals. In contrast to serial communication digital I/O uses several inputs and outputs in parallel, usually 16. Digital inputs and outputs in NeuroCheck are configured in the Device Manager dialog.

Digital camera

In the NeuroCheck Software the term "Digital Camera" refers to the digital camera types that directly transfer their data using a bus, without requiring a frame grabber:

- FireWire® (IEEE 1394 a/b) cameras
- GigaBit-Ethernet cameras

Driver DLL

A DLL (Dynamic Link Library) is a special type of Windows module that supplies other programs with specific functions. NeuroCheck uses DLL's for controlling special hardware devices, e.g. image processing boards. The functions of the DLL translate NeuroCheck's commands into the corresponding commands understood by the individual hardware device. NeuroCheck also supports the use of plug-in DLLs containing user-defined image processing functionality.

Features

In NeuroCheck all kinds of information describing an object, respectively a region of interest. Examples are simple scalar values computed by function Compute features or visual information provided by function Resample regions of interest. The features are stored together with the regions of interest. Subsequent function can use the features by accessing the corresponding list of regions of interest in the data pool.

Field Bus

A field bus system is used to connect various stations or work cells distributed along a production line or throughout a whole factory area. Well-known field bus systems are PROFI-Bus, Interbus-S and CAN-Bus.

Filtering

Filtering changes the contents of a gray value image. A convolution kernel is moved across the image. From the gray values of all pixels inside the kernel a new gray value is computed according to a filter algorithm. The resulting gray level is inserted into the result image. Filtering can be used for smoothing images, enhancing and extracting edges, broadening or thinning image structures etc.

Final Action

The **final action** is a special individual check which is always executed after all the other checks. The Final action is typically used for communication activities necessary after all image processing has been completed. The Final action is added to the end of a check routine by choosing Edit ▸ New ▸ Final Action.

FireWire®

FireWire® cameras are digital cameras that can be connected to a PC and controlled through a IEEE1394 bus. Thus no image processing board is needed.

Frame Grabber

Synonym for image processing boards.

Gauging

Function group "Gauging" comprises functions for the determination of complex geometrical measurements of objects, derivation of new measurements and checking of allowances.

Geometrical Description

Geometrical structure, representing an object. Every object has the geometrical descriptions "Point" and "Contour", denoting its center of gravity and its silhouette. In addition, model geometries can be computed, approximating pieces of a contour with ideal geometrical structures, like lines and circles. Geometrical descriptions are mostly used by gauging functions.

Gigabit Ethernet

Gigabit Ethernet is a PC network technology. Gigabit Ethernet digital cameras use this technology to transfer image data at a speed of up to 1000 MBit/s.

Gradient

The gradient is the difference between gray levels of neighboring image pixels. Gradients give information about edges and textures in the image.

Gray Level Image

A gray level image contains the complete information about the brightness of a scene, but no color information. Usually images with brightness values in the range of [0..255] are used.

Gray Level

Numerical value for the brightness of an image pixel, usually in the range of [0..255].

Group of ROIs

Regions of interest can be combined in groups, provided the grouping feature of check function Define ROIs has been activated. Group membership of an ROI depends on the group number of the manually defined ROI, which acted as starting point for the creation of the respective ROI. It can, however, be altered subsequently. For a detailed explanation of the concept of groups refer to section "Groups of ROIs".

Hardware key (dongle)

Every NeuroCheck license comes with a hardware key (a so called "dongle"). The license level as well as the NeuroCheck version are encoded in the hardware key. Without a hardware key, NeuroCheck runs as a Demo version. Two different types of hardware keys are available: one for the parallel interface and one for a USB interface of a PC. For more details, please refer to the hardware key section.

Histogram

A histogram depicts the distribution of numerical values. In image processing, histogram frequently denotes the distribution of gray levels in an image. In this case it consists of 256 entries, corresponding to the number of distinct gray levels. Each entry represents the frequency of the corresponding gray level in the image. From the histogram, thresholds for the segmentation of image objects can be determined.

Image Acquisition

The complete process of transferring an image from the camera via an image processing board to computer memory for further processing.

Image Analysis

Function group "Analysis" comprises functions for searching, measuring, counting and classifying objects within an image.

Image Preprocessing

Processing steps that create a result image basically similar to the source image (same size, similar contents), but different in certain aspects. Examples are filters for noise reduction or edge enhancement, look-up tables for contrast enhancement etc.

Image Processing Board

Also called "Frame Grabber". Interface between computer and video camera. The image is read from the camera, digitized and stored until it is transferred to the main program for further processing.

Image Section

In NeuroCheck, image section (in contrast to a region of interest) denotes that part of a bitmap file or a camera image transferred to memory by the function Transfer image.

Input Data

Data that a check function needs to perform its processing task. The input data objects have to be supplied by the preceding check functions of the same individual check to be accessible inside the data pool.

Kernel

A kernel determines the image points from the neighborhood of a pixel to be used for filtering. Depending on the shape of the operator, specific structures in the image can be enhanced or suppressed, e.g. horizontal and vertical edges.

Look-up Table

A look-up table assigns to every possible gray level a new gray level. This allows every conceivable brightness transformation, even if it cannot be expressed as a mathematical formula. Look-up tables can e.g. be used for contrast enhancement.

Mean Value Filter

The mean value filter replaces the gray level of each pixel by the average gray level in a neighborhood of the pixel, defined by the convolution kernel. It removes small image degradations at the cost of edge blurring.

Measurement List

Data object, containing results of a measuring operation, as computed by function Gauge ROIs.

Median Filter

A median filter sorts the gray levels in a neighborhood of each pixel defined by the convolution kernel. The gray level of this pixel is replaced by the gray level in the medium rank position. It removes small image degradations more effectively than the mean value filter with almost no perceptible edge blurring, but at the cost of higher processing time due to the sort operation.

Model Geometry

Approximation of a contour line or the connection between the centers of gravity of objects within a group by an ideal geometrical structure, like a circle or straight line. Can be used as geometrical description for gauging.

Morphology

Using morphological operations specific image structures can be directly influenced. Basic morphological operations are erosion and dilation. Eroding results in a thinning of bright structures, dilation widens them.

Multilayer Perceptron

Widely used type of neural network, consisting of one input, one output and at least one hidden layer. In practice, a single hidden layer proves to be sufficient for almost any application. The multilayer perceptrons have always one hidden layer in NeuroCheck.

Neural Network

Neural networks are used as classifiers. Their specific properties are the capability to learn from examples and to recognize a pattern even if it is only similar to the training patterns, not exactly identical. This robustness makes them ideally suited to recognition tasks like reading letters.

OEM

"Original Equipment Manufacturer"; here: someone who uses NeuroCheck as development system for his own image processing applications.

OK/not OK

The final result of inspecting a work piece can be "OK", meaning that all requirements of the check have been fulfilled, or "not OK". In automatic mode NeuroCheck can display this result on the screen and report it to the PLC or host computer via digital I/O, serial interface or other communication channels.

Object

Here used in the sense of "image" object. Objects are extracted from an image for further processing by segmentation. Every object is regarded as a new region of interest by NeuroCheck, so that every operation applicable to regions of interest can also be used for objects. For example, functions Create ROIs by thresholding and Template matching can be used to search for objects within an object.

Output Data

Results created by a check function All result data is collected in the data pool, where it is available to subsequent check functions.

Parallel (asynchronous) Image Capture

Parallel image capture requires a digital camera or frame grabber board capable of processing external trigger signals. After activating the board by function Capture image in parallel it waits for the trigger signal to capture an image. The check routine can run in parallel during that time. Check routine and image capture are again synchronized the next time this check function is executed.

Password Protection

A check routine, which has been protected using function Password Protection cannot be changed, unless the correct password is entered first.

Pattern

A **pattern** in NeuroCheck is a collection of features describing a single object i.e. one region of interest. Such a pattern can be used to classify a region of interest.

Pattern File

File containing a collection of patterns used for training a classifier.

Pattern Set

A **pattern set** in NeuroCheck is stored in a training data file and is used for training a classifier.

Payload Data

In the context of result value output from NeuroCheck to files or other interfaces, payload data denotes the values computed by check functions, like features, measurements etc.

Persistent Serialization

Data are permanently saved to a medium and can be read again and displayed at a program restart (e.g. data in a database).

Pixel

Individual point within an image.

Plug-In Check Function

 Plug-in check functions are part of a Plug-In DLL. If you add such a check function to a check, it will be indicated with the above symbol in manual mode.

Adding a plug-in check function is only possible if you own a "Premium" license. Otherwise you cannot acces the Check Function Group "Plug-In" in the check function dialog.

Plug-in check functions are not part of the standard NeuroCheck installation but were developed by external companies or people based on the specifications for the NeuroCheck plug-in interface. Plug-in check functions can be integrated seamlessly into the NeuroCheck development and run-time environment.

Plug-In DLL

A plug-in DLL is a .NET Assembly meeting the NeuroCheck plug-in- interface specifications. It serves to enhance NeuroCheck with user-defined image processing functionality and custom menu entries. The section License Levels of NeuroCheck" describes the possibilities of using plug-in-DLLs in the different NeuroCheck editions.

Position Adjustment

Function Position ROIs is able to adjust the position of regions of interest automatically, using correction values computed by function Determine position of ROIs according to position and orientation of a reference object.

Record

In NeuroCheck a record consists of all values written to file during one execution of a single check function including additional control information. Compare with the data packet transmitted via serial interface.

Reference Image

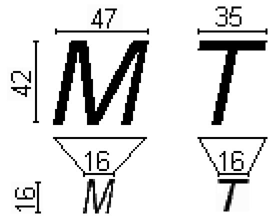
Image with markings at conspicuous locations, manually set using the Reference images dialog. NeuroCheck stores such images together with the check routine to be used for adjusting camera positions in automatic operation.

Region Of Interest (ROI)

Computations of certain check functions. in NeuroCheck are restricted to regions of interest (ROI). Such a region can be defined manually or constructed automatically, e.g. from object contours. It can therefore be of any shape. A ROI always represents an object in NeuroCheck which can be measured and classified etc.; on the other hand every object found by NeuroCheck is immediately available to subsequent check functions as a new region of interest.

Resampled Image

A resampled image in NeuroCheck is created by converting a region of interest to a prescribed number of pixels. This is necessary if image data is to be used directly by a classifier, because classifiers expect a constant number of input values (in this case: pixel gray levels). The example shows that the letter 'M' is wider than the letter 'T'. Therefore the enclosing rectangles of the two letters comprise a different number of pixels. After resampling the two letters consist of the same number of pixels, so the resampled images can be used for classification.



Usually the number of pixels will be smaller after resampling, although this is not necessary. If it is the case, the original image section is subdivided into fields of equal size. Each field is represented by one pixel in the resampled image. The brightness of this pixel is either equal to the average brightness of all the pixels inside the corresponding field of the source image or equal to the brightness of the center point of the original field. The first method is more accurate, but slower.

Result Records

In the context of result value output from NeuroCheck to files or other interfaces, result records denote the "O.K."/"not O.K." results of individual checks and the check routine.

Search Ray

A line across the image along which NeuroCheck searches for edges or objects.

Segmentation

The process of separating objects from the image background is called segmentation. After successful segmentation the objects can be described and processed independent of the image.

Serial Communication

In contrast to digital I/O serial communication does not use parallel inputs and outputs. The signals are instead transmitted as continuous stream of bits.

Serial Protocol

The serial protocol determines how data is encoded when transmitted via serial communication. This comprises for example start and stop signals, error checking bytes etc. The protocol is realized by the device driver used for serial communication and is explained in the documentation provided with the device driver.

Signature

A signature is a one-dimensional representation of a two-dimensional property of an object. For example, function Compute Curvature creates a vector containing a curvature value for every pixel on the contour of an object. Another example is the internal representation of an object contour by the distance of the contour points to the object's center of gravity used internally by function Determine Position to compute the orientation using the polar distance method.

Start Action

The **start action** is a special individual check which is always executed before all the other checks. A typical use for start action is fast capturing of several images, which can then be stored in the image tray and processed later by individual checks. A Start action is added to the beginning of a check routine by choosing Edit ▸ New ▸ Start Action.

Stopwatch

The stopwatch window is opened by choosing **Stopwatch** from the Tools menu. It shows the execution times in milliseconds, either of check functions or the entire check.

Target Point

The pixel within the limits of a filter kernel, to which the result of a filter operation is assigned. In most cases the target point will be located in the center of the kernel.

Threshold

A threshold is used to compute a binary image from a gray level image. Each gray value above the threshold is set to the gray value 255, representing "White", all others to 0 for "Black".

Thresholding

Converting a gray level image to a binary image. Each gray value above the threshold is set to the gray value 255, representing "White", all others to 0 for "Black".

Training Data

Training data files (extension '*.TD') are created Training Data Wizard opened from within the parameter dialog of function Classify ROIs. Such a file contains a collection of regions of interest together with feature values and class information to be used for training a classifier.

Trigger-(Signal)

This is a trigger signal, e.g. in combination with a camera or control system (PLC).