NEURO CHECK

Industrial Vision Systems

Version 5.1
for Microsoft® Windows®
2000 / XP / Vista / 7

Training Course

Version 5.1 of Nov 02, 2010

Contents

1 Introduction 1-1
1.1 Terms 1-2
1.1.1 Application Areas 1-2
1.1.2 Check Routine Hierarchy 1-3
1.1.3 Modes of Operation 1-4
1.2 The Intro Screen 1-6
1.3 The NeuroCheck Window 1-8
1.3.1 After Program Start 1-8
1.3.2 Version Information 1-8
1.3.3 Online Help 1-10
1.4 Configuration of the Image Acquisition 1-14

2 Working with Check Routines 2-1
2.1 The Check Routine Wizard 2-2
2.2 Editing the Check Routine 2-6
2.3 Outlook: Automatic Mode 2-8
2.4 Manual Configuration of Check Routines 2-10
2.5 Camera Images in a Check Routine 2-16
2.6 Check Routine Documentation 2-20

3 Automatic Mode 3-1
3.1 Configuration of Digital I/O for Communication 3-2
3.2 Remote Control 3-4
3.3 Automatic Inspection Procedure 3-6
3.4 Automatic Check with Type Change 3-8
3.5 Output Window Configuration 3-10
3.6 Function Key configuration 3-14

4 Presence Verification 4-1
4.1 Determining the Presence of an Object 4-2
4.2 Presence Verification Using Object Properties 4-8
4.3 Inspection Statistics 4-16

5 Gauging 5-1
5.1 Measuring Object Distances 5-2
5.2 Change of Target Values in Automatic Mode 5-8
5.3 Gauging Parallel Edges 5-12

5.4 Output of Measurements to a File 5-20
5.5 Using Metrical Units 5-26

6 Position Adjustment 6-1
6.1 Compensating for Translations 6-2
6.2 Compensating for Rotations 6-10
6.3 Transmitting Position Parameters 6-14

7 Optical Character Recognition 7-1
7.1 Creating Training Data 7-2
7.2 Creating a Classifier 7-12
7.3 Target Value Configuration 7-16

8 Template Matching 8-1
8.1 Searching Objects with Template Matching 8-2
8.2 Optimization of Template Matching 8-8

9 Advanced Topics 9-1
9.1 Diagnostic Methods 9-2
9.1.1 Configuration of a debugging screen layout 9-2
9.1.2 Activate saving of error images 9-4
9.2 Troubleshooting 9-6
9.2.1 Software Trouble Report 9-6
9.2.2 Log File 9-6
9.3 Configuration of the Image Acquisition for frame grabbers 9-8
9.4 Configuration of the Serial Interface 9-10
9.5 Data Output using the Serial Interface 9-12

10 Appendix: Images 10-1
10.1 Bar Codes 10-2
10.2 Demonstration Bitmap 10-2
10.3 Circuit board for template matching 10-3
10.4 Disks 10-4

1 Introduction

Welcome to the NeuroCheck training course. This course will familiarize you with NeuroCheck, the user-friendly interactive machine vision software for Windows, and the broad range of automated visual inspection tasks that can be solved with NeuroCheck. The course can be used for self-education or as a supporting document in a seminar.

In order to work through this course you need an installation of NeuroCheck 5.0 or higher in Premium, Professional, Training or Demo version. All images used throughout the course are included as bitmap files, so that you do not need a camera. The sample files for this training course are part of a complete installation of NeuroCheck 5.x. Alternatively, you can install them using the **NeuroCheck Setup** command from the NeuroCheck start menu. The files will be copied into a subdirectory `Training` of the NeuroCheck installation directory.

The training course should be worked through in the given order, because explanations of basic NeuroCheck usage will become sparser, whereas the image processing problems will become more demanding. Obviously, it is impossible to cover the complete functionality of NeuroCheck in this training course. For more information and details please refer to the User Manual.

The NeuroCheck software including the complete documentation

1.1 Terms

First of all, we will briefly explain a few expressions used in NeuroCheck to give you some idea of what we mean by these words. The necessarily short explanations will become clearer over the course of the training, but they will serve as a starting point to the world of solving machine vision problems with NeuroCheck.

1.1.1 Application Areas

NeuroCheck can be used to solve a variety of visual inspection problems. These can be roughly categorized into the following application areas.

Bar Code Identification

Everyone knows the rows of vertical lines on almost any item of daily use, from books and magazines over candy bars to rental cars. NeuroCheck is able to identify all standard types of bar codes and convert them into plain text.

Presence Verification

In many applications the presence of a prescribed number of objects has to be verified, ranging from the simple counting of connector pins or tablets in a blister to completeness checks of whole assembly groups, where several different kinds of objects have to be found and distinguished according to properties like size and shape.

Object and Image Comparison

This application area is concerned with verifying the correspondence between the current image and a reference image. For mark or print quality inspections a complete image is typically used as the reference; other applications may store a whole set of objects as reference, complete with various feature values. The latter type of application is similar to a presence verification with the added difficulties that the objects have to be laid out in a specific arrangement and may have very different characteristics, making it very complicated to check their presence by simple measuring and screening operations.

Gauging

Whereas in presence verification only a more or less precise correspondence of object properties like size and orientation to predefined values is required, the purpose of a gauging application is to determine geometrical measurements of objects with high precision – including relationships between two or more objects, like distances, angles etc. With NeuroCheck you cannot only determine standard measurements of objects like dimensions, orientation angles, area or radii. In addition it is possible to model an object by an ideal geometrical structure (e.g. a straight line as an approximation of one edge of an object or a circle as an ideal representation of a more or less circular object), determine properties of this model geometry and compare these properties to those of the original object. Thus you can determine the roundness of an object, i.e. its deviation from the ideal circle, or the straightness of an edge. Furthermore it is possible to compute such a model geometry for a complete group

of objects. Thus you can check whether pins on an IC deviate from their correct location or whether four or more bore holes on a work piece are precisely positioned on a circle. NeuroCheck uses special interpolation algorithms to perform gauging with subpixel precision.

Pattern and Optical Character Recognition

Pattern recognition denotes the identification of arbitrary patterns using measurements as well as their overall appearance. It is often used as a tool for presence verification to distinguish between valid and invalid objects. Optical character recognition (OCR) is a special case of pattern recognition and one of the most demanding applications in digital image processing. NeuroCheck uses neural networks as classifiers for the identification of characters (or other patterns). In contrast to most neural network packages NeuroCheck concentrates on the problem to be solved instead of neural network issues of interest only to the researcher. Every step, from the generation of the input data for the classifier to the training data collection and the actual network training is visually guided, thus making neural network applications easier than ever.

Robot Guidance

The term “Robot guidance” 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.

1.1.2 Check Routine Hierarchy

Image processing solutions in NeuroCheck follow a simple hierarchy, clearly visualized by a tree structure in the NeuroCheck window:

Check Routine

The check routine comprises the complete solution to one inspection problem. It can consist of one or more individual checks. On the check routine level parameters like the identification number (for check routine switching by remote control), password protection, reference images, check execution in automatic mode and data output settings are stored.

Check

A check, short for “individual check” or “single check”, consists of a sequence of check functions. It solves a specific task within the inspection problem. A common structuring criterion are camera images: every inspection to be performed on one camera image is done in one check.

Checks do not exchange data. Their sequence is therefore unimportant, unless the synchronization with the manufacturing process or the communication with the PLC demand a certain sequence.

Check Function

A check function fulfills a particular image processing operation, like an image filtering or an object search. Check functions are carried out in a user-defined sequence. Each check function can access all data objects created by preceding check functions. Check functions belong to different function groups which are explained below.

1.1.3 Modes of Operation

NeuroCheck knows several different modes of operation.

Manual Mode

In manual mode you can interactively configure check routines in a point-and-click fashion with all the amenities of a modern Windows-based application. Also you can configure hardware and software settings.

Live Mode

In live mode the whole NeuroCheck window is used to display the live image from a selected camera to let you adjust focus and exposure of the camera without the need for an external video monitor.

Automatic Mode

In automatic mode NeuroCheck will be remote-controlled by digital I/O or serial interface to perform automated visual inspections. It can write data like measurements, identified bar codes or recognized characters to a file or transmit it via serial lines for further evaluation in a networked quality management system.

1.2 The Intro Screen

Double clicking on the NeuroCheck icon on the windows desktop will start the program.
After start-up, NeuroCheck displays a Web-like intro screen which gives convenient access to the most important functions.
The graphical buttons correspond to the following commands:

Open

Opens a file select dialog for selecting an existing check routine file. The check routine is then read from the file and displayed in the manual mode view in NeuroCheck.

New

Creates a new empty check routine and displays it in manual mode view.

Wizard

Calls the Check Routine Wizard which will assist you with the creation of new check routines for typical application areas.

Demo

Loads a demo check routine and displays it in manual mode. This check routine demonstrates sample inspections from various application areas using bitmap images.

First Steps

Switches to a tutorial explaining the first steps with NeuroCheck to give novice users a quick start into the program.

Info

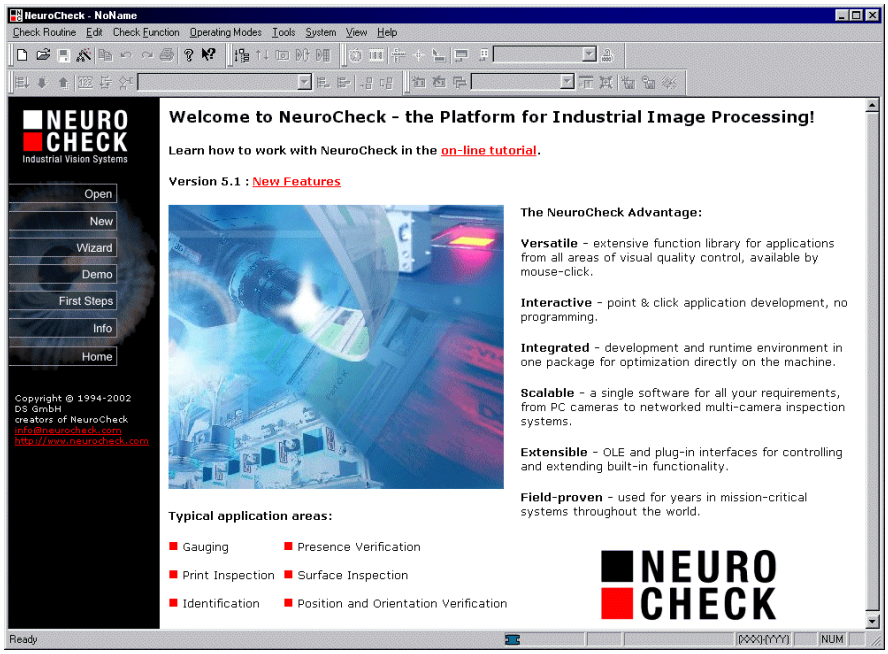
Displays a dialog with information about the NeuroCheck version currently running.

Home

Connects you to the NeuroCheck home page in the World Wide Web.

New Features

Shows an overview of all new features regarding to previous versions including changes in the currently installed Service Pack [SP].



NeuroCheck starts with a Web-like intro screen.

1.3 The NeuroCheck Window

Problem

Now that you have started NeuroCheck for the first time you want to find out what to do with the elements of its user interface.

Result

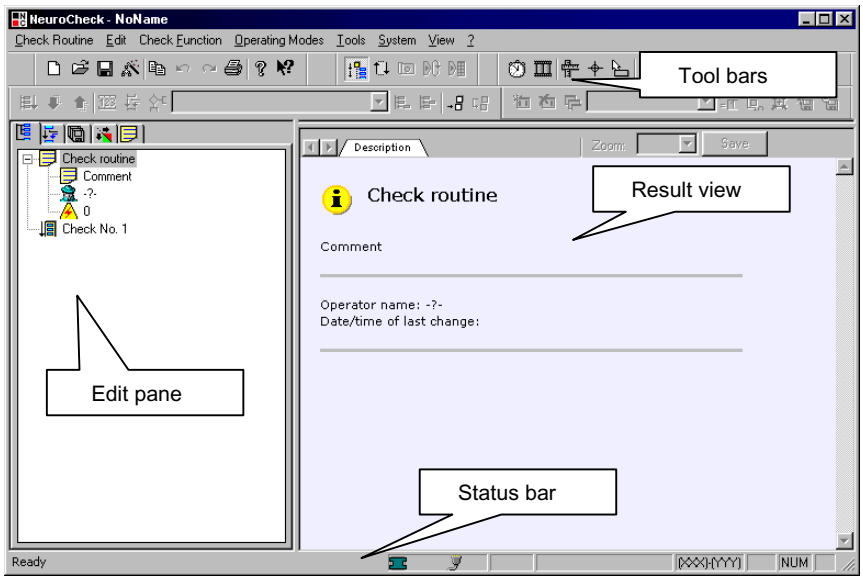
After you worked through this section, you are able to determine the version of NeuroCheck you are using, access information in the online help system and configure the NeuroCheck user interface.

1.3.1 After Program Start

After clicking the **New** button on the intro screen you will see the NeuroCheck window in its default configuration in manual mode. Below the menu bar you see the standard NeuroCheck toolbars, which will be covered in detail later in this section. The main window area is divided into the edit pane on the left, featuring tabbed edit pages, and the result image display on the right. The bottom border of the window contains the status bar, which keeps you informed about the current state of the program.

1.3.2 Version Information

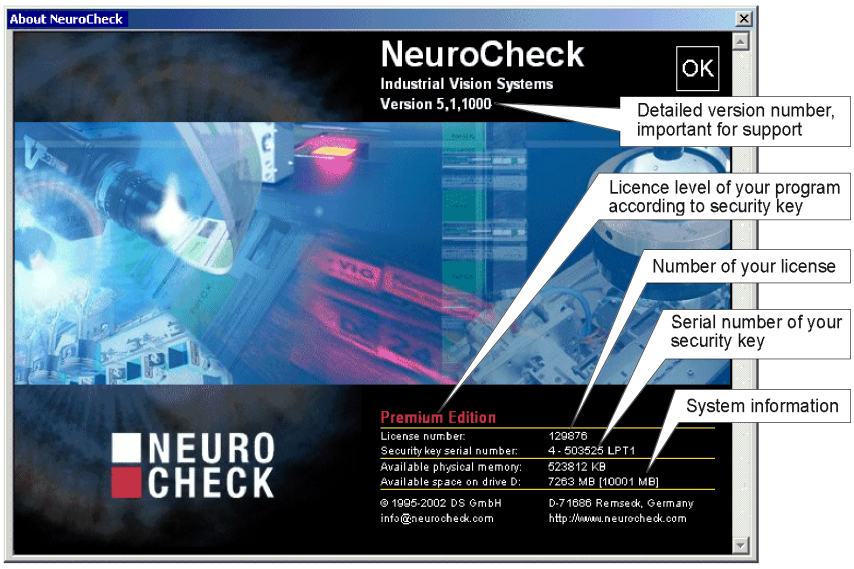
Choosing **About NeuroCheck** from the **Help** menu opens a dialog box with information about the version of NeuroCheck you are using. The same dialog box is displayed upon clicking the **Info** button on the intro screen. The dialog box contains the exact program version, license numbers of program and security key and the license level. If you remove the security key from your PC and restart NeuroCheck you will notice that the displayed license level has been changed to Demo Version. The demo version of NeuroCheck restricts the use of special hardware like cameras, digital I/O boards, serial interface etc. and does not allow for the automatic execution of check routines in an industrial environment.



The NeuroCheck window after clicking **New** on the intro screen. The main window area in manual mode consists of the edit pane with tabbed edit pages on the left and the result view on the right.

A small icon in the status bar indicates the presence of a security key.

The **Help** Menu gives access to the help system and the program information dialog.



You can reach the program information dialog by choosing **About NeuroCheck** from the **Help** menu.

1.3.3 Online Help

Problem

You want to look up information about individual control elements or functions in NeuroCheck and do not have this information available in printed form.

Result

You can access information about any topic in the online help system of NeuroCheck, have it displayed and printed.

Help system contents table

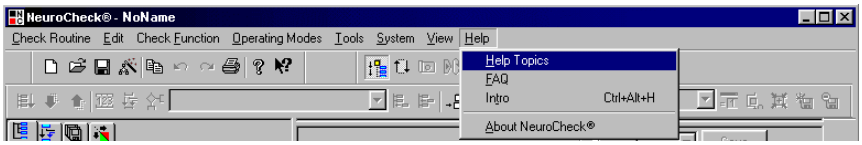
The menu item **Help ▸ Help Topics** opens the contents table of the NeuroCheck help system. A dialog box containing a number of closed books appears. Double clicking the books with the left mouse button opens the books and closes them again. Some of them contain "sub books", others the actual help topics, indicated by a ?-icon. Double clicking such an entry displays a help window with information on this topic.

Help system index

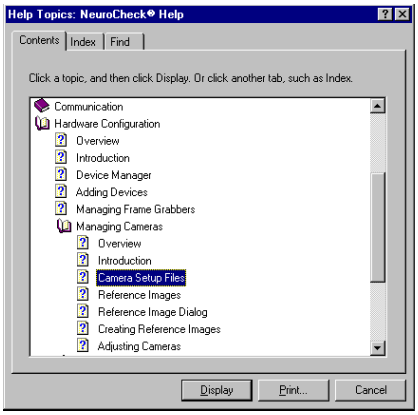
The dialog box with the help system contents table consists of several so-called property pages. You can switch between these pages by clicking the tabs on top. On the second page you can search an alphabetic index. Choosing the **Display** button or double clicking a keyword opens the help window with the pertaining topic. If the keyword occurs in more than one topic, a dialog box appears first, allowing you to choose from the topic titles.
The Windows help system remembers which property page you used most recently Choosing the menu item **Help ▸ Help Topics** again immediately displays this page.

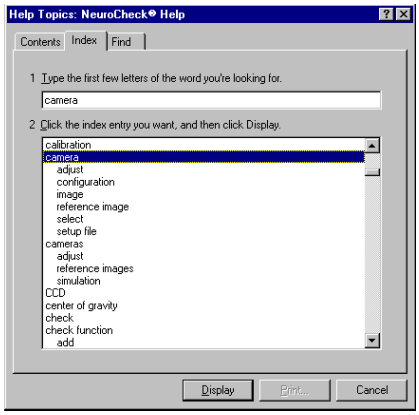
Cross references and definitions

A mouse click on underlined words in the help window takes you to related topics with further explanations. The **Back** button takes you back to topics visited since last opening the help window. Dotted underlining indicates definitions. Clicking such a word opens a popup window with a short explanation. The popup vanishes upon your next mouse click.
Clicking the **See Also** link in the non-scrolling header part of the help window again opens a dialog box listing related topics corresponding to those found via keyword search. The other links in the header area of the help window take you one or two levels higher up in the help system hierarchy.

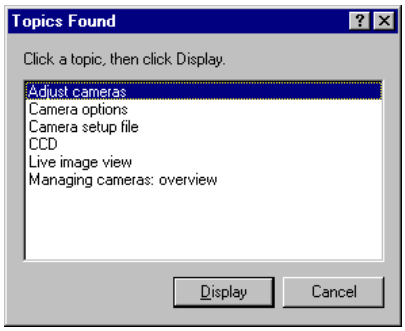


The menu item **Help Topics** displays the table of contents of the NeuroCheck help system.

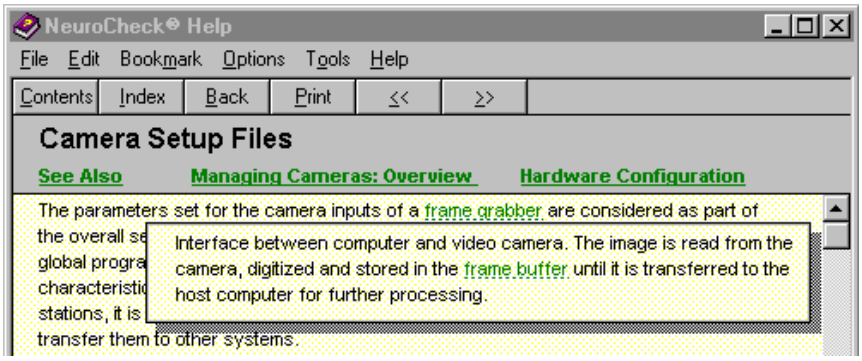




The individual books of the help system can be opened and closed by double clicking. A single click on the tabs on top of the pages switches between the contents and index pages.



If there are several entries containing a keyword, this select box appears first, otherwise the help window is displayed immediately.



Clicking dotted words displays a short explanation in a popup window. Underlining indicates cross references, which you can follow by a single mouse click.

Context sensitive help for control elements

The topics in the help system are arranged in a systematic order, not unlike a printed manual, which is very helpful for retrieving basic information and surveys. If you simply want to know the meaning of a menu item, the effect of an icon in an icon bar or the function of a control element in a dialog box, accessing this information via contents or index tab can be quite time-consuming. To get information on an icon you would have to call the overview of all icon bars, search for the correct icon bar and the look up the correct icon. Fortunately, there are much simpler ways to access this information.

F1 Menu item: Open the menu, move the mouse on the menu item so that it appears selected, but do not click it. Press the F1 key. The help window appears immediately. For menu items with sub menus, indicated by the ▸ icon, this works on the lowest level only (because this is the only level actually calling program functions).

Icon bar: click the contextual help icon depicted on the left, then the icon you need information on. Again the help window appears immediately.

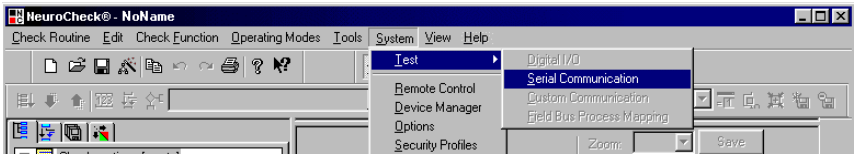
Dialog boxes: most dialog boxes also contain a contextual help icon. Choose the menu item **Settings** in the **System ▸ Software** menu. In the dialog box click the question mark icon in the title bar, then any control element. Information about this control element appears in a popup window.

Printing help topics

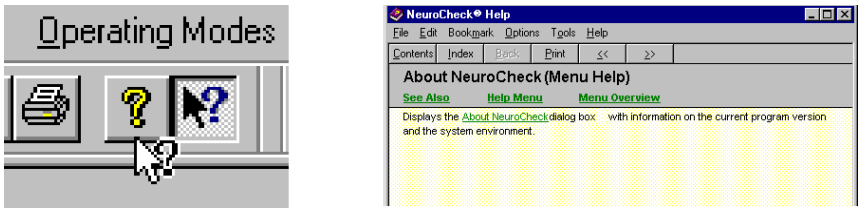
If you need information from the help system in printed form you can have single topics as well as whole sections of the help system printed. Open the contents tab of the help system, select the topic or book to be printed and choose the **Print** button. The topic or all topics within the selected book will be printed.

Copying help information

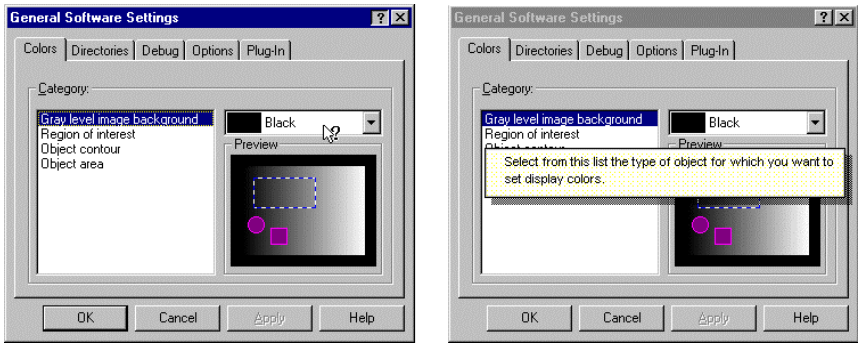
You can transfer text from single help topics via the Windows™ clipboard to other applications. Select the region of text to be copied with the mouse and choose **Copy** from the **Edit** menu. If the help window does not have a menu bar, you will find this menu item in the context menu reached by clicking the help window with the right mouse button.



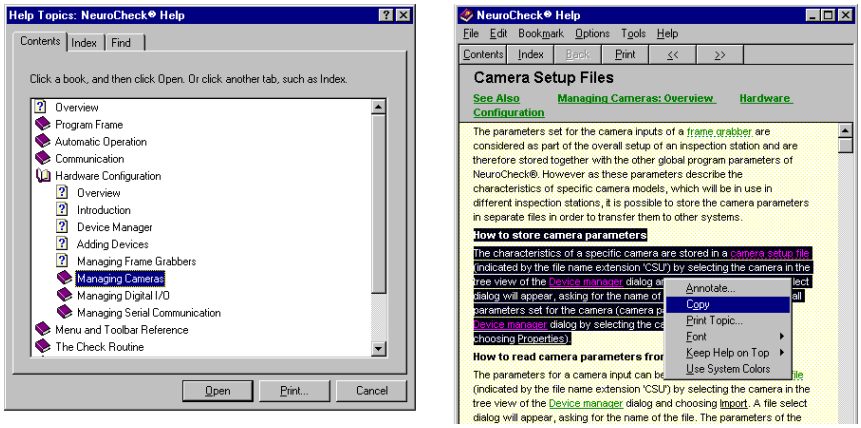
Pressing the F1 key opens the help window directly from a menu item.



The contextual help cursor opens the help window directly from an icon in an icon bar.



With the ? icon you can access help information on every control element in a dialog box.



You can print whole books from the help system and transfer text from single topics to other applications via the clipboard.

1.4 Configuration of the Image Acquisition

Problem

You want to capture images with NeuroCheck for the first time and check whether focus and exposure on your camera are set correctly.

Result

You are able to configure a FireWire camera for use with NeuroCheck and know how to have a camera image displayed.

Solution

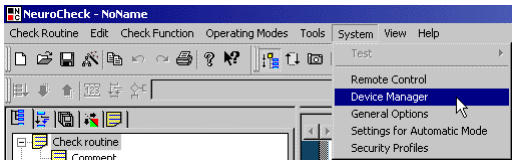
❶ After installing the hardware in your computer you have to set up the low lever driver delivered by the manufacturer. Then you have to register the new hardware in NeuroCheck. Connect all FireWire cameras to the computer. From the **System** menu choose **Device Manager**. The Device Manager is the central dialog to configure and register new hardware in NeuroCheck. Choose **New** in the **Device Manager** dialog box.

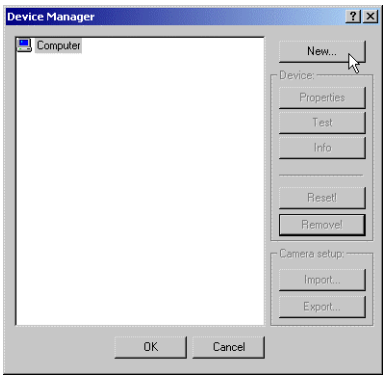
❷ The hardware wizard will guide you through the setup process. On the first page of the hardware wizard select **IEEE1394 (FireWire) Camera.** On the second page select the specific FireWire driver.

❸ As soon as you confirm your selection on the final page of the hardware wizard, the system searches for connected FireWire cameras. When a camera has been found and its reactions has been verified, it is entered into the device tree. The device identification of the camera is automatically detected and set as a default name for the new camera.

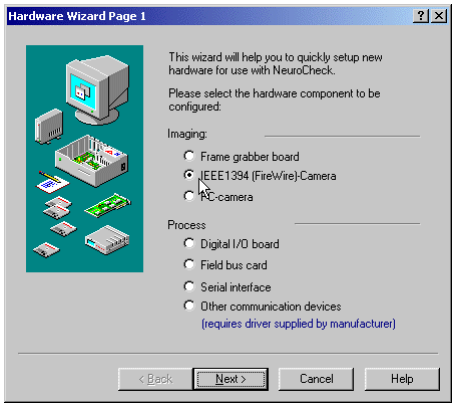
The camera name is used consistently throughout the software. You can change the name here by in-place editing: click the camera designation once with the left mouse button, then a second time (make a pause, do not double-click). Then enter a new name, for example “Top View”. From now on NeuroCheck will know this camera under this particular name.

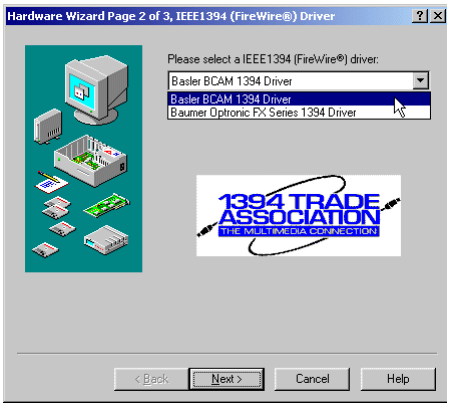
❹ When the camera or multiple cameras has been detected, the icon and the menu item for live image view become accessible. In addition the status bar displays a camera icon next to the security key icon.



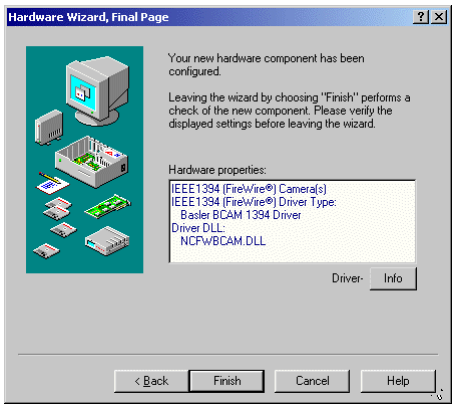


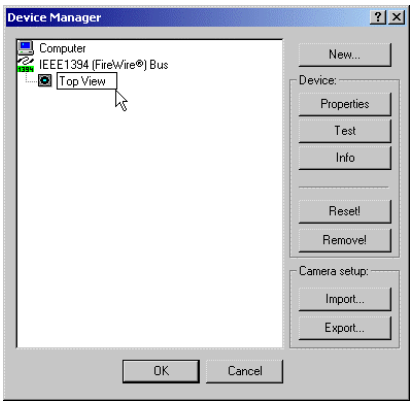
❶ From the **System** menu choose **Device Manager** to configure your hardware. Choose **New** in the Device Manager dialog.



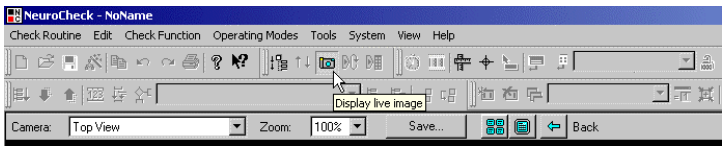


❷ The hardware wizard will guide you through the setup process.





❸ After confirming your selection NeuroCheck will scan for connected FireWire cameras and will enter the device IDs in the device list. You can change this name by in-place editing.



❹ After configuration functions for live image view become available

Live image display

❺ 

Choose the **Live image** icon. NeuroCheck switches to live mode, using the whole client area of its window for displaying the current camera image.

The top border of the live window contains several control elements. From the leftmost list box you can select the camera by the names given in the Device Manager dialog.

The second list box sets the zoom level. A zoom level of 100% gives the best representation for focusing, but in order to get an overall impression of what the camera sees, especially if it is a high-resolution camera, lower zoom levels might be more appropriate.

Clicking the **Full Screen** symbol will use the complete screen for displaying the live image. Click again on this symbol to switch back to normal mode.

If you have connected more than one camera you can switch to a split screen mode by clicking the **Split Screen** symbol. This mode display up to four camera images at once. This is very useful to set up the cameras and the lighting because shadows and image quality is visible for multiple views at the same time.

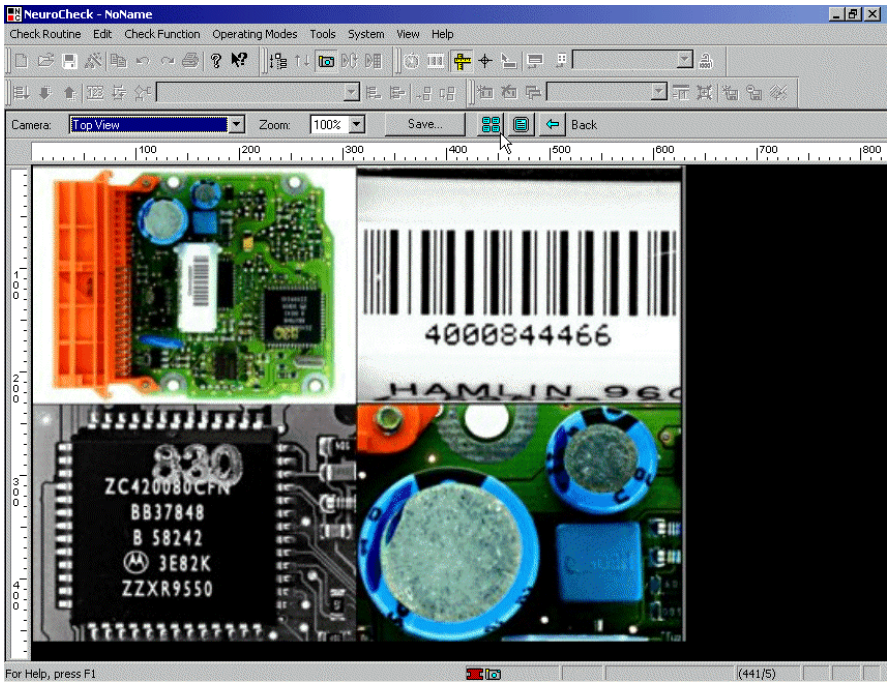
The **Save** button finally lets you save the image as a Windows bitmap file. Optionally NeuroCheck can insert date and time as well as a descriptive string into the lower right corner of the stored image to help you documenting your applications. Images stored via the **Save** button can be viewed and edited in most Windows applications.

Please note that during live image display NeuroCheck continuously loads full frame images from the camera into main memory. Depending on the system this may consume a considerable amount of processing power and slow down other activities like reactions to user commands.

❻ If you want to change the camera properties like video resolution, gain or shutter you can use the Device Manager, too. Open den **Device Manger** from menu **System** once more.

Select the camera for which you want to change the parameters and choose **Properties**. The opened dialog gets its information about the parameters directly from the FireWire camera. That means that the range of values for all parameters in the dialog corresponds to the supported values of the connected camera.

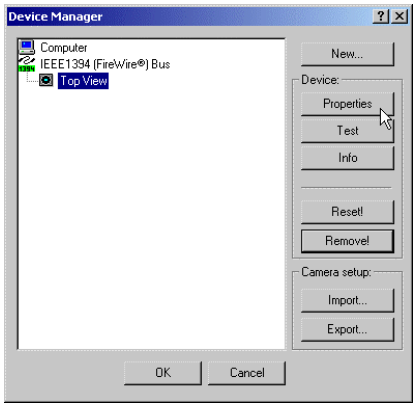
Switch to page **Scalar Properties**. If you increase the value of the parameter **Brightness** using the slider your image will become lighter. Choosing **New!** will display a small preview of the current camera image to check the camera settings. Leave this dialog with **OK**. After closing the Device Manager by clicking OK, the changes will take effect and can be seen in the live image.

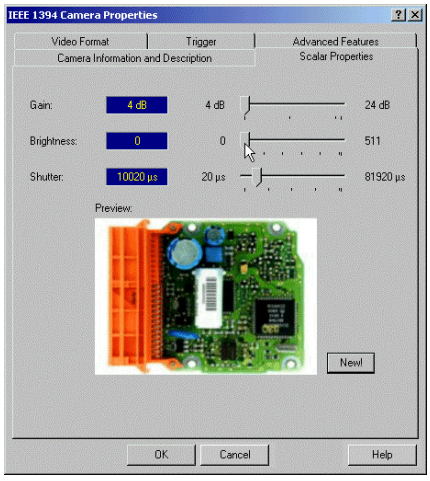


❺

Live image view

If more than one camera is connected and registered in the system, the display of up to four camera images at the same time is possible.





❻

Select the camera for which you want to change the parameters and choose **Properties**.

Change the settings for **Brightness** on page **Scalar Properties**.

2 Working with Check Routines

In this chapter you will learn

- how to create check routines and manage them as documents,
- how to use the check routine wizard to create a check routine for reading a bar code,
- how to use the elements of the check routine window,
- how to run a check routine in automatic mode (you will get an outlook on the automatic mode in NeuroCheck).
- how to build a check routine for bar code identification step by step,
- how to use camera images in a check routine for “live” inspections,
- how to document your application in parallel to working with NeuroCheck, e.g. to create a feasibility study.

Check of bar codes on toothpaste tubes

2.1 The Check Routine Wizard

Problem

You want to create your first check routine to see NeuroCheck at work.

Result

You are able to configure a simple image processing application using the check routine wizard and execute this check routine.

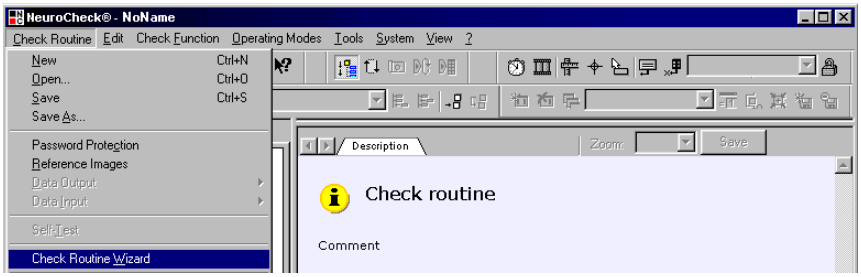
Solution

❶ 
From the **Check Routine** menu choose **Check Routine Wizard** or click the **Wizard** button on the intro screen. The first page of the check routine wizard appears, giving some introductory information. The **Next** button takes you to the second page.

❷ On the second page of the check routine wizard you can choose between several basic types of image processing applications. Select the **Bar code identification** button, because this is the simplest type of application as far as the configuration in NeuroCheck is concerned. Then again click the **Next** button. The third page contains some options for dealing with images of low quality, which can be ignored for the time being. Simply choose **Next** once more.

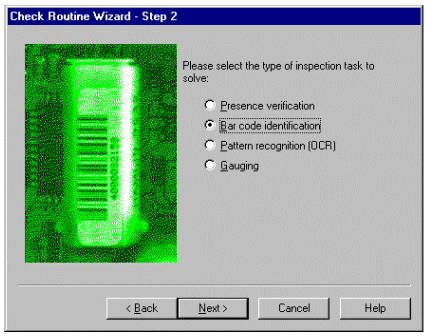
❸ On the fourth page activate the **Image processing on bitmap file only** check box. Thus we can guarantee identical conditions for this first try at an image processing application. Now choose the **Finish** button.

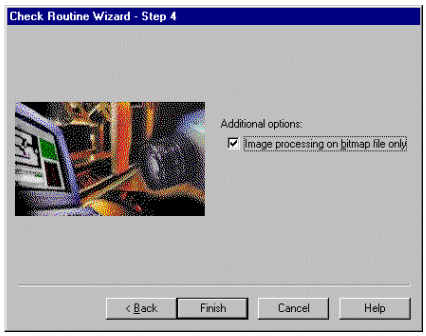
❹ A small dialog box reports that the check routine wizard creates and saves a new check routine. Shortly afterwards the edit pane in the left portion of the NeuroCheck window displays several icons and lines of text. The last one reads **Check No. 1**. Click on the ⊞ icon in front of the text to have NeuroCheck display the structure of the check. Several new lines appear, containing the **check functions.** Each of those fulfills a specific image processing task.



❶ 

From the **Check Routine** menu choose **Check Routine Wizard**.



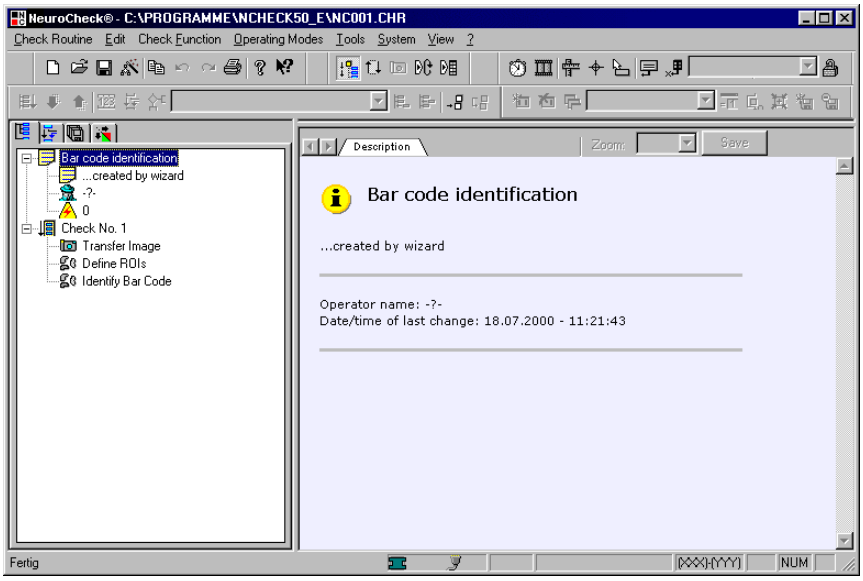


❷

On page 2 select **Bar code identification** as the type of application

❸

On page 4 select image processing from bitmap file only.



❹

The check routine wizard creates a new check routine and displays it. Clicking on the ⊞ icon in front of the **Check No. 1** line opens the check so that its structure becomes visible.

❺ 

Activate the first check function, Transfer image, by clicking it with the left mouse button. From the **Check Function** menu choose **Execute** or click the depicted icon from the edit bar. The right window pane now displays a gray level image containing a bar code. If the right window pane is not wide enough, you can move the border separating the window panes by dragging it with the left mouse button held down, until you can see the complete bar code. Alternatively you can change the zoom level from the list box in the top window border.

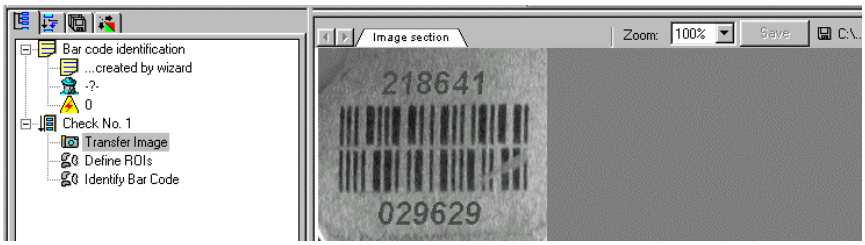
❻ 

Now click this icon in the edit bar or choose **Next** from the **Check Function** menu. In the left pane of the window the next check function, Define ROIs, is selected. In the right window pane a rectangle appears, crossing the bar code. This is the region used by NeuroCheck to search for the bar code.

❼ 

Execute the next check function, Identify bar code. in the same way. In the right window pane a very short list appears, consisting of one line with the bar code read. When you move the mouse pointer to the top window border above the **Zoom** list box it changes to a separation symbol. Hold the left mouse button down and drag the mouse downwards. The right window pane is separated and you can see the image of the bar code together with the search region in the upper pane, the list in the lower pane.

Each of the two panes has its own tabs for selecting the view displayed in this pane. When the pane displaying the image is the active pane (i.e. the last one clicked with either mouse button), its **Save** button becomes available, allowing you to store the image information in a Windows bitmap file. This pane also has a context menu, opened by clicking the pane with the right mouse button, allowing you to copy the image to the Windows clipboard.

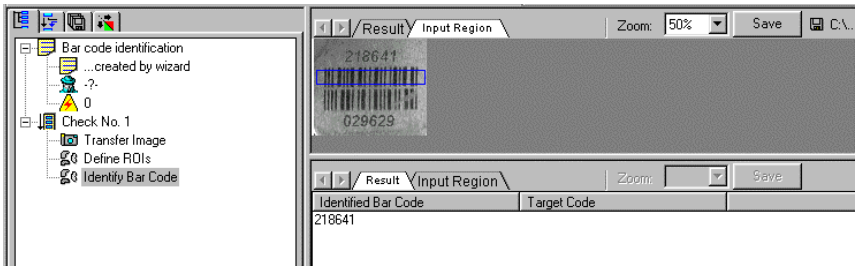


❺ 

Result of executing the first check function, Transfer image.

❻ 

Result of executing the second check function, Define ROIs. NeuroCheck searches the displayed region for the bar code.



❼ 

In the split right window pane NeuroCheck displays the search region as well as the identified bar code.

2.2 Editing the Check Routine

Problem

On one work piece two bar codes have to be checked that are so far apart that the field of view of one camera does not give sufficient resolution to identify the bar codes safely. Apart from the position of the bar codes the inspection tasks are identical and you do not want to configure everything from scratch for the second bar code.

Result

You can use the check routine tree view, copy and delete checks.

Solution

❶ 

Select the line **Check No. 1** in the check routine tree view by clicking it with the left mouse button. Then click the depicted icon from the edit bar or choose **Copy Check** from the **Edit** menu. Below the first check of the check routine an identical sequence of check functions appears. You have just created a second check, which gives you a good start for configuring the second solution, because of the similarities of the two inspection tasks.

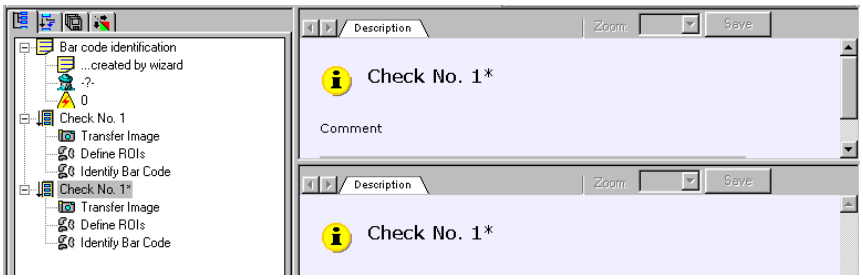
Note the asterisk which has been added to the title of the second check to avoid having two checks with identical names.

❷ 

Click the minus icon in front of the first check with the left mouse button. The check functions vanish, only the title line of the check remains. The minus icon changes into a plus icon to indicate that something more is hidden beneath this line. Clicking the plus icon causes the check functions of the first check to reappear. The same effect can be produced by double clicking the title line of the check.

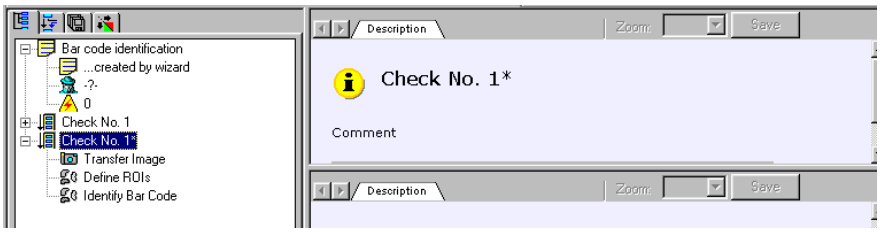
❸ Later, the two checks will have to fulfill different tasks, therefore they should have appropriate names. Click the title line of the first check with the right mouse button and choose **Properties** from the context menu. In the dialog box which is then displayed you can enter a name for the check, a comment, and a detailed description, optionally in HTML format. Confirm with **OK**. The texts entered here are displayed in the right window pane when the check is selected in the tree view. They provide important information for the developer of the check routine as well as the end-user.

❹ Check routines with several checks will be treated in detail later. For now you simply delete this check again. Click the title line of the second check with the right mouse button and choose **Delete** from the context menu, reverting the check routine to its original form.



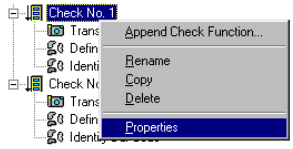
❶

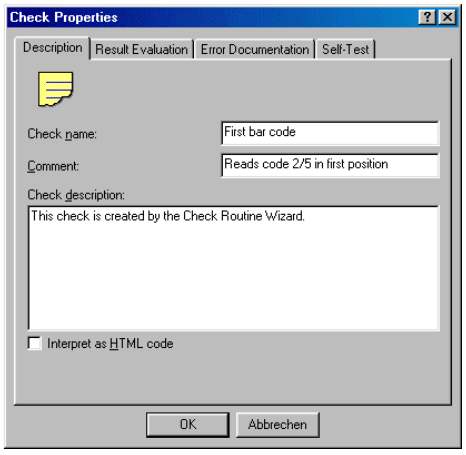
Check routine after copying the first check.



❷ ⊟⊞

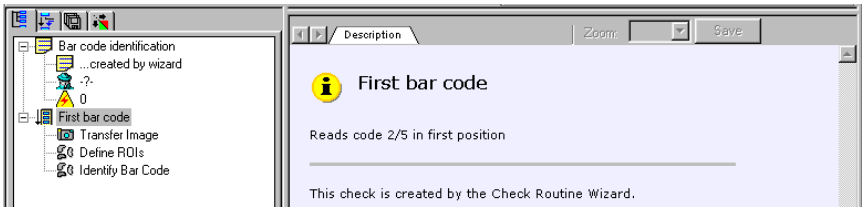
Check routine with the first check “folded shut”. Click the ⊟/⊞-icons to open and close checks.





❸

Descriptions and explanations of checks will be useful for the solution of similar problems later.



❹

Check routine after deleting the second check. Note the description of the first (and only) check in the right window pane.

2.3 Outlook: Automatic Mode

This section is intended to give you a brief impression how easily you can configure a visual inspection program, which can be integrated into an automatic production process. Of course, a real application needs some additional settings to enable NeuroCheck to communicate with your PLC or master computer, but most of these are global settings done only once. Automatic execution of a finished check routine is indeed as simple as it is shown here.

❶ From the **System** menu choose **Remote Control**. On the **Input Signals** page activate the **Start check** signal by clicking the box in front of the signal designation. The **Source** column should read **Manual [<ENTER>]**, otherwise you will have to press the **Change** button and select **Manual** from the **Select Signal Source** dialog box. Confirm with **OK**.

❷ 
Choose **Automatic** from the **Operating Modes** menu or click the corresponding icon. NeuroCheck switches to the automatic mode screen and displays three windows. The display area of the center window read **Waiting for start signal**. Now press the **Return** key. Briefly, the display will read **Running**, then revert to **Waiting for start signal**. The main area of the center window will display several lines, informing you about the executed checks. The leftmost window will appear green and signal **Part O.K.**, whereas the rightmost window will remain blank.

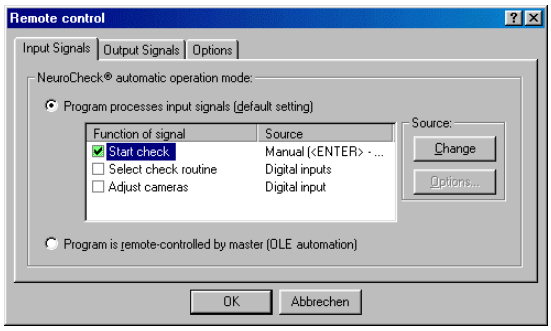
You have just executed an automatic check routine for the first time. In a real-world application some outside instance, like a PLC, would send a signal equivalent to pressing the **Return** key, and NeuroCheck would send some output back, but there is no essential difference to the procedure just executed.

❸ 
Now choose **Configure Automatic Screen** from the **Operating Modes** menu or click the corresponding icon. The layout of the NeuroCheck window in automatic mode is completely configurable and as a first step we will tell NeuroCheck what to display in the rightmost window. Click the window with the right mouse button and choose **Parameters** from the context menu.

In the tree view of the **Check Function Result Window Parameters** dialog activate the checkbox of function Identify bar code and confirm with **OK**.

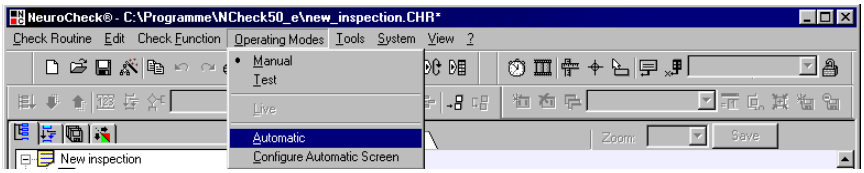
❹ 
Switch back to automatic mode and press the **Return** key. The check routine is executed again, but this time the rightmost window will display the identified bar code.

Chapter 3, **Automatic Mode**, will treat the automatic mode and its settings in detail.



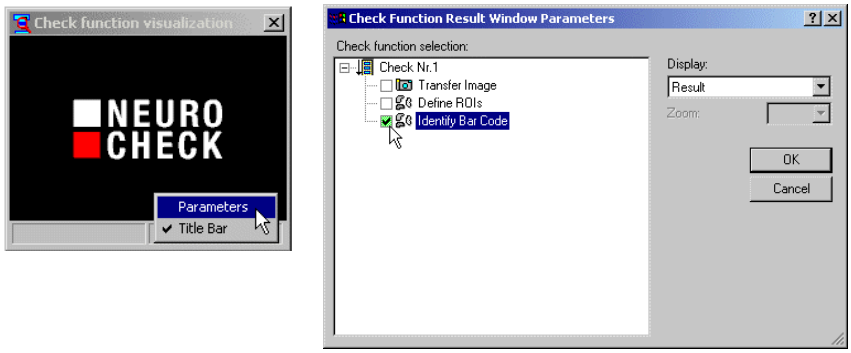
❶

From the **System** menu choose **Remote Control**. Activate the **Start check** signal.



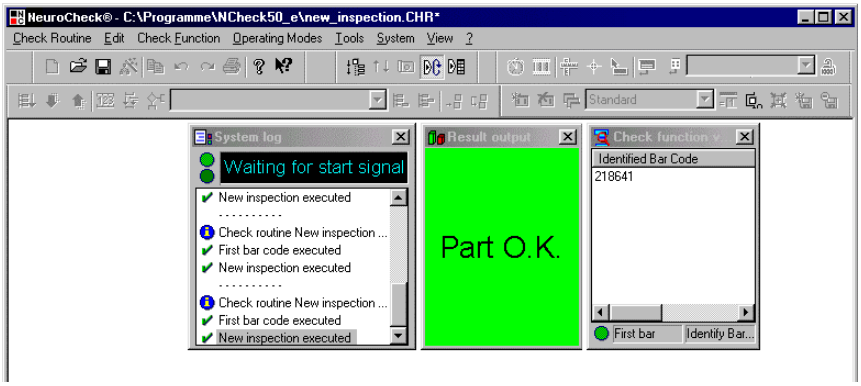
❷

Switch to automatic mode and press the **Return** key to execute the check routine.



❸

In Screen Configuration Mode choose **Parameters** from the context menu of the check function result window and activate the checkbox of function Identify bar code.



❹

Switch back to automatic mode and press the **Return** key. The rightmost window now displays the result of function Identify bar code.

2.4 Manual Configuration of Check Routines

Problem

The preconfigured check routines supplied by the check routine wizard will create a basic framework for your inspection tasks. Normally you will have to build the check routines step by step.

Result

You are able to build a check routine step by step and set the parameters of the individual check functions.

Solution

Before starting to solve new inspection problems, you will learn the basic operations for configuring new check routines by manually rebuilding the check routine from the previous chapter.

❶ 

First create a new check routine. Enter “Bar code identification” in the title line (**Description** in the **Check Routine Properties** dialog), “Manually configured check routine” in the second line (**Additional explanations** in the **Check Routine Properties** dialog). Save the check routine, e.g. as “Bar code manually”.

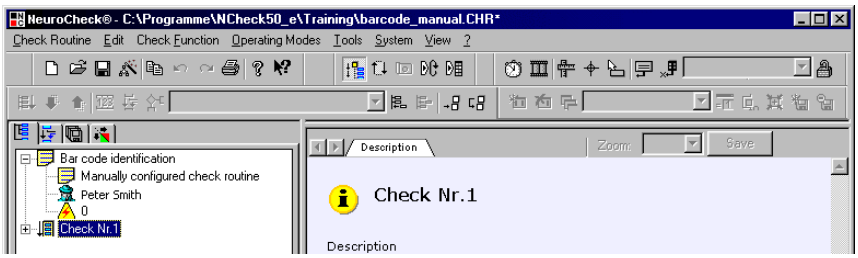
❷ 

The first check function in the check routine created by the wizard was Transfer image. This will be the first check function in our new check routine, too. Choose **New ▸ Append Check Function** from the **Edit** menu or the context menu opened by clicking the title line of the check (reading **Check No. 1**) with the right mouse button. Alternatively you can click the icon on the left. The dialog box **Select New Check Function** will appear.

Click the **Acquisition** button. You will now see all check functions pertaining to capturing and loading images. Select function Transfer image with a mouse click and confirm with **OK**. The dialog box is closed and the first check function appears in the check routine window.

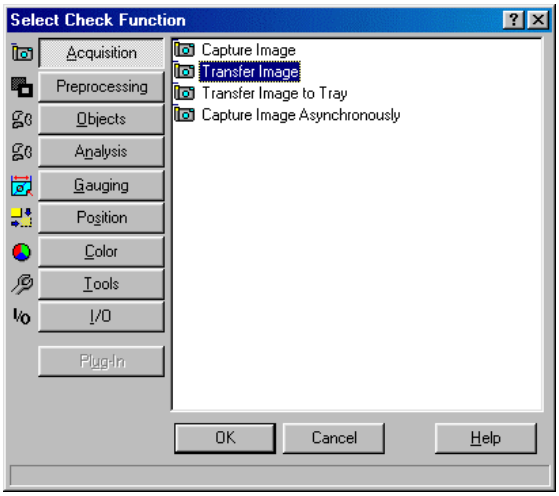
❸ 

The next step is to tell the check function where to take the image from. The check function is already selected, therefore you can open the parameter dialog immediately. From the **Check Function** menu or the context menu of the check function itself choose **Parameters** or click the corresponding icon from the edit bar. Depending on your system configuration a message box may inform you that image access failed (because the program tried to capture from a not connected camera). You can safely confirm this message box. The parameter dialog of function Transfer image will appear.



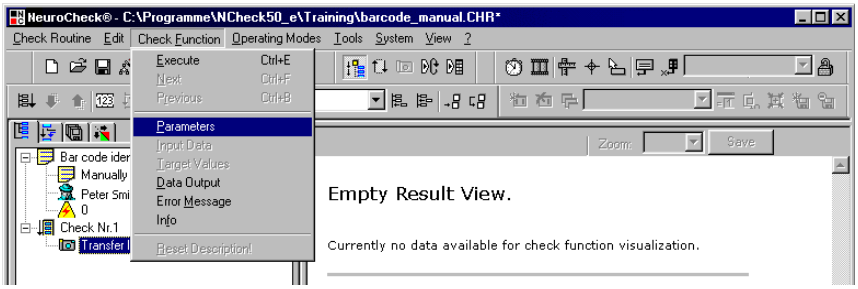
❶ 

Create a new check routine, set the title and comment string and save the check routine.



❷ 

After choosing **New ▸ Append Check Function** from the **Edit** menu or the context menu of the check itself, click the **Acquisition** button and select function Transfer image.



❸ 

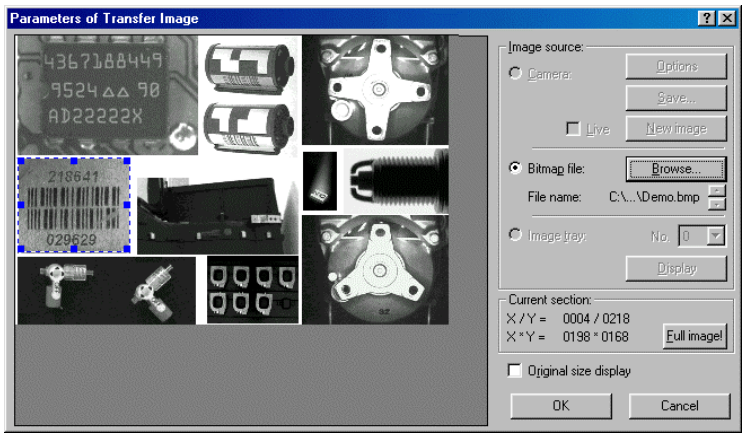
Open the parameter dialog of the check function.

❹ In this section we will still use a bitmap file as the image source. Select the check box **Bitmap file**. NeuroCheck in its default configuration will automatically load the demonstration bitmap included with the program. If the demonstration bitmap is not displayed, you can click the **Browse** button to load the `demo.bmp` file from the `Examples` directory below the NeuroCheck installation directory.

Click the image section drawn in blue with the left mouse button, hold the mouse button down and draw the image section over the bar code. Drag the markings in the corners and the borders of the image section keeping the mouse button down to change the size of the image section. A reliable bar code identification requires some space to the left and right of the bar code, so do not set the image section too small. Confirm with **OK**.

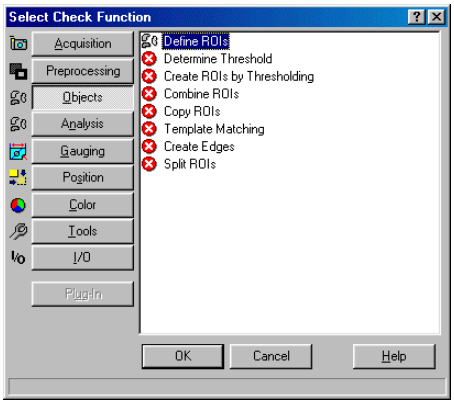
❺  Append check function Define ROIs to the check as in step ❷. You will find this check function by clicking the **Objects** button because regions of interest are the central data objects in NeuroCheck.

❻  Again some parameters have to be adjusted. Open the parameter dialog as in step ❸. Draw a rectangle across the bar code, leaving some space to the left and right of the bar code. If you do not find the right size and position for the rectangle immediately, you can click the **Select** button and modify the rectangle in the same way as the image section in step ❹.



❹

Select the demonstration bitmap included with the program and draw the image section around the bar code, providing enough space to the left and right of the code.



❺ 

After choosing **New ▸ Append Check Function** from the **Edit** menu or the context menu of the check itself, you can select function Define ROIs from the **Objects** page.

❻ 

Open the parameter dialog as in step ❸ and draw a rectangular region of interest, covering the bar code with enough space on the left and right side.

❼ 

Append check function Identify bar code to the check. You will find this function by clicking the **Analysis** button because it analyzes the image within the defined region of interest, trying to identify a bar code and comparing it to a target value.

Open the parameter dialog in the usual way. From the list box **Bar code type** select type **Code 2/5 Interleaved**. Deactivate the check box **Execute check** because one does not always know in advance, whether a particular bar code contains a check sum or not. Set the number of characters to 6 and leave all other settings on their default values.

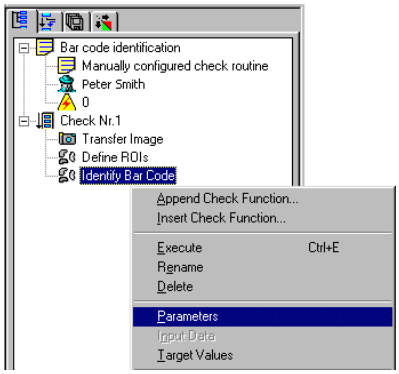
❽ 

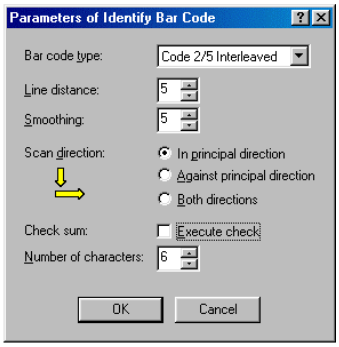
As an evaluation function, Identify bar code has a target value dialog, opened by choosing **Target Values** from the **Check Function** menu or the context menu of the check function in the edit pane. The corresponding edit bar icon symbolizes the decision between good and faulty parts made by the function. Activate the target value comparison in the dialog and enter the string `218641` as the target value.

❾ 

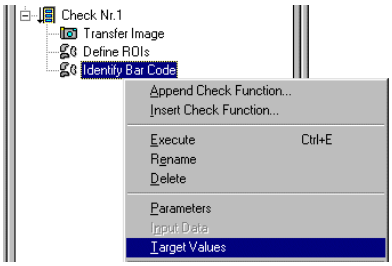
After leaving the parameter dialog with **OK** and executing the check routine yields the same result as the check routine created automatically by the check routine wizard.

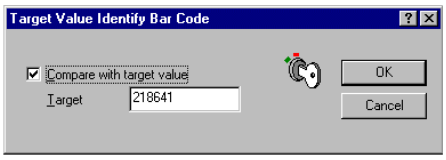
The check routine just created is exactly equivalent to the check routine from the previous chapter in that it also uses a bitmap file as its image source. Checking bar codes on different test pieces would require creating a bitmap file for every check piece. The following section explains how a real application captures the bar code directly with a camera.



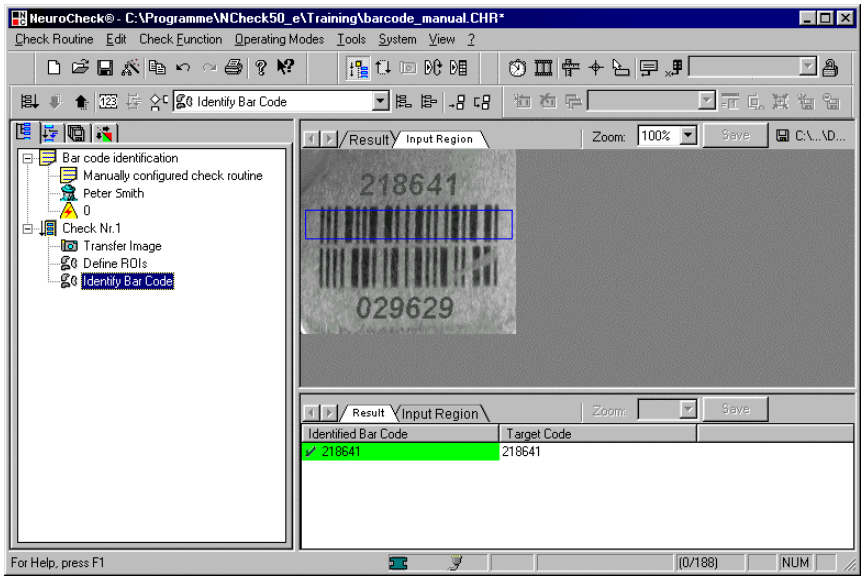


❼ On page **Evaluation** select check function Identify bar code and set its parameters.





❽ Open the target values dialog of the function and activate target value comparison.



❾ The check routine yields the same result as the one from the check routine wizard.

2.5 Camera Images in a Check Routine

Problem

A real-world application integrated into a production process has to inspect the current test piece, i.e. the image source has to be a camera, not a bitmap file.

Result

You can configure a check to use a camera image.

Solution

❶ 

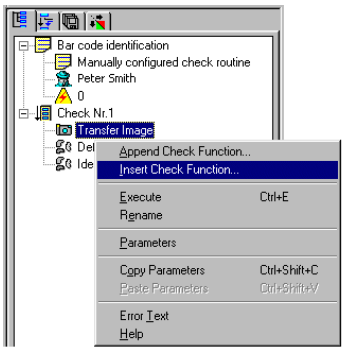
Select the first check function, Transfer image, with the left mouse button. From the **Edit ▸ New** submenu or from the context menu of the check function choose **Insert Check Function** or click the depicted icon from the edit bar to insert a check function immediately above the selected check function. Click the **Acquisition** button and select check function Capture image.

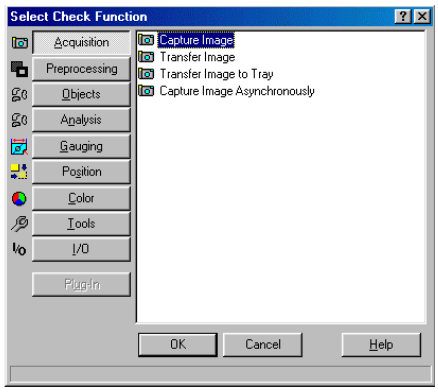
❷ 

Open the parameter dialog of the function (this is possible only, if hardware for image capturing has been installed before and was configured correctly). Select the right camera.

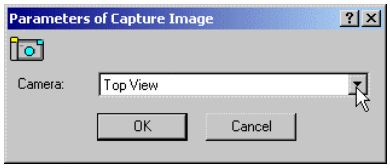
❸ 

Open the parameter dialog of function Transfer image. Activate the **Camera** button to have the function use the image data recorded by function Capture image. If there are several cameras connected to your system, you will have to select the right camera by pressing the **Options** button. The **New image** button triggers capturing a new image from the selected camera. You can use the bar codes on page **10-1** of this training manual as samples. Activating the **Original size display** check box makes precise adjustments to the location and size of the image section easier.

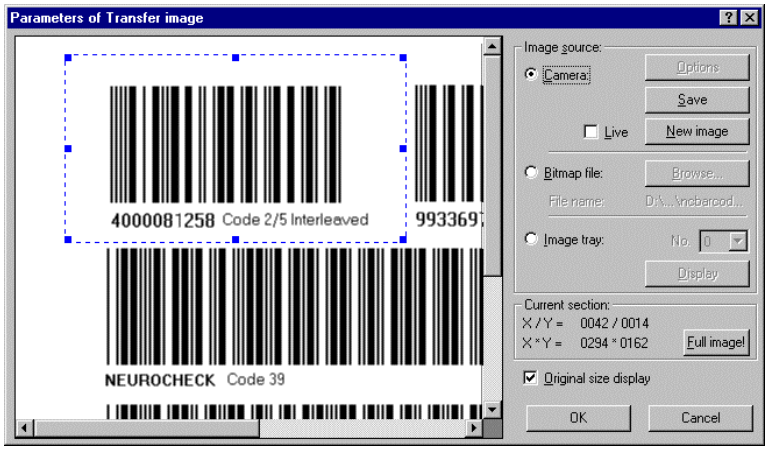




❶ From the **Edit ▸ New** submenu or from the context menu of the check function choose **Insert Check Function** to add function Capture image prior to the first check function.



❷ Select the camera to be used for capturing the image.



❸ Set the image source of function Transfer image to camera and adapt the image section. Precise adjustments are easier when **Original size display** is activated.

❹ 

The dimensions of the image section and the position of the bar code within the image section may have changed after switching to the camera image, requiring the region of interest to be adjusted. Open the parameter dialog of function Define ROIs. If the image section is smaller than before, NeuroCheck may inform you that a region of interest has become invalid, because it does not fit into the smaller image section. In this case you have to enter the region of interest anew. Otherwise you can click the **Select** button, move the region and change its size until it covers the bar code again.

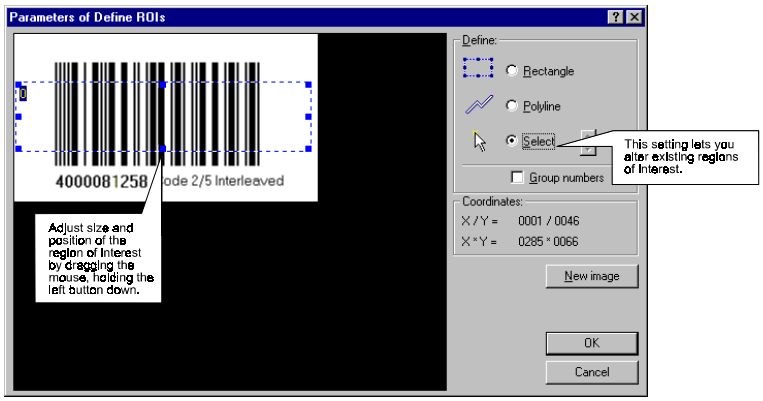
❺ 

If you use one of the bar codes on top of page 10-1, NeuroCheck reads the code without further parameter changes except setting the number of characters to 10. For other bar codes you may have to select a different type of bar code in the parameter dialog of function Identify bar code. The codes are of course reported as 'not O.K.', because they do not correspond to the value set in the target value dialog of function Identify bar code.

❻ 

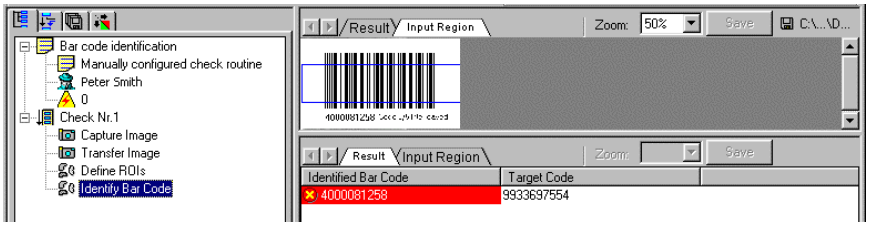
The upper two bar codes on the bar code page are of the same type. When the check routine is executed in automatic mode, you can move the page between two inspection runs, so that the bar codes can be read alternately (for convenience you should construct some kind of stop mark indicating how far you have to move the page for the bar codes to appear in the same place in the camera image).

If you change the target values of function Identify bar code to one the two codes, e.g. `9933697554`, you will get alternating 'O.K.' and 'not O.K.' results.



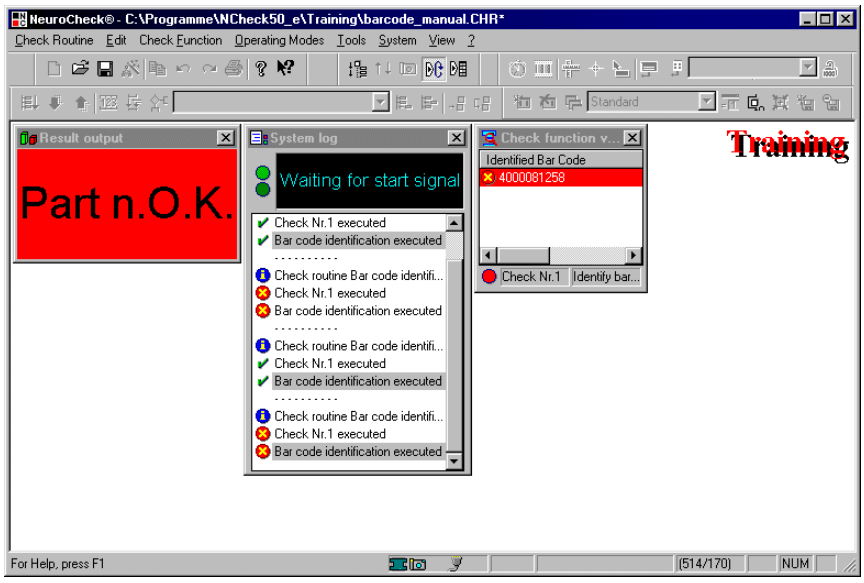
❹ 

Adjust the region of interest to the altered image section.



❺ 

NeuroCheck reads this code without further parameter changes.



❻ 

NeuroCheck in automatic mode. The `4000081258` code is classified as ‘not O.K.’, because it does not equal the target value of function Identify bar code.

2.6 Check Routine Documentation

Objective

You want to document your bar code identification check routine, and use this documentation for a feasibility study.

Result

You can create a documentation for the behavior of your check routine while working with NeuroCheck, export it into the XML format understood by Internet Explorer, and know how to determine the execution time of a NeuroCheck check routine.

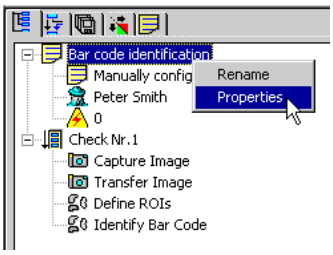
Solution

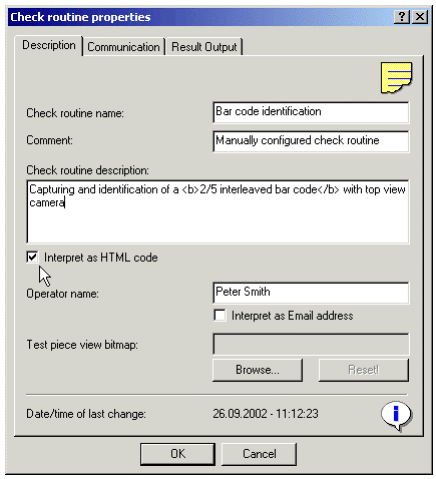
We assume that you found a suitable camera and lighting arrangement and configured a check routine for reading the bar code. Now you have to document the setup and the check routine. This is done directly in the check routine. You can enter descriptions for the entire check routine as well as for individual checks and check functions.

❶ In **Structure View** select the item **Bar Code Identification** representing the check routine. Right-click the highlighted area and select **Properties** from the context menu. Among other fields, you will find the text box **Check Routine Description** on the **Description** tab. Enter your description of the check routine's tasks and operation. Example: „Capturing and identification of a 2/5 interleaved bar code with top view camera." The two strings in angle brackets are HTML commands marking the beginning and end of an area in bold font style. You can use complex HTML formatting such as type fonts, lists, pictures or even hyperlinks to web sites. Use standard HTML commands or import source code from an HTML editor. In addition check the **Interpret as HTML code** box. Enter into **Operator name** the name of the person responsible for this check routine; in this example: "Peter Smith".

❷ Confirm with **OK**. The result display on the right hand side now shows the text you entered (formatted as HTML). To open a context menu, right-click the result display window. You can select a command such as **Print** to print each HTML page.

❸ In the same manner you can document individual checks and check functions. In Structure View select the check function Identify bar code, open the context menu and choose **Properties**. As before, you can enter a text on the **Description** tab. For example: "Reads the 2/5i bar code in the region of interest. 10 characters are read and compared to the target value 9933697554." Confirm with **OK**.

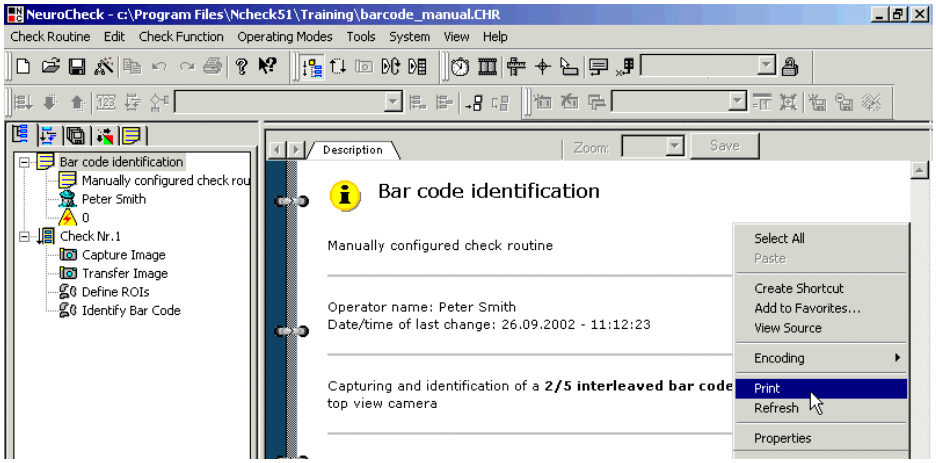




❶

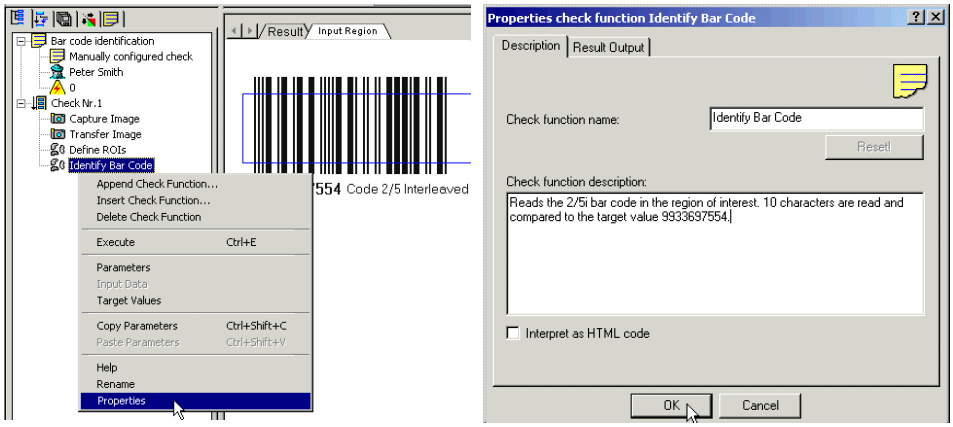
Open the context menu and choose **Properties**.

If you are using HTML commands, do not forget to activate the corresponding check box.



❷

The result display shows the description as an HTML page. This page can be printed directly from NeuroCheck.



❸

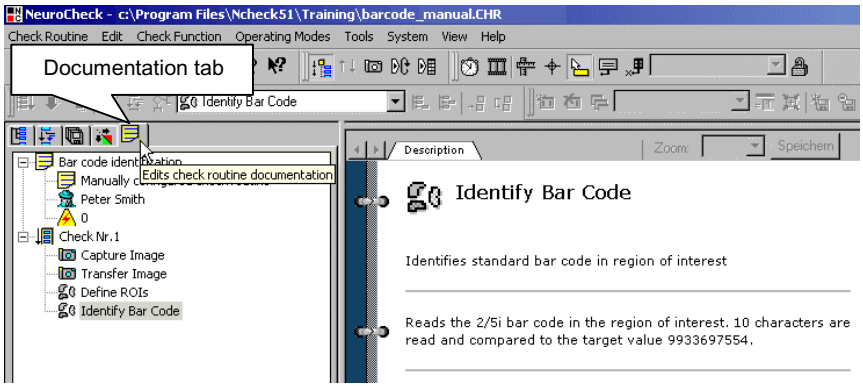
Check functions can also be documented and commented using the HTML description.

❹ To display the documentation of check functions, switch to the fifth tab page **Documentation** in Structure View. The text you entered is then displayed on the right side.

❺ Furthermore, you can export all information such as check routine structure, check function parameters plus the description to an XML file for documentation purposes. In the **Check Routine** menu select **Export ▸ as XML...**. A file selection dialog is displayed, in which you can determine the directory and the name for the output file. The check routine's name is the default. The file automatically receives the extension `.chrx` (Check Routine in XML format). For all information with all parameter settings to be included, select the default setting **Detailed** for the **Visualization Mode**.

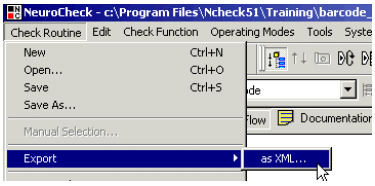
Select **Save** to complete the export.

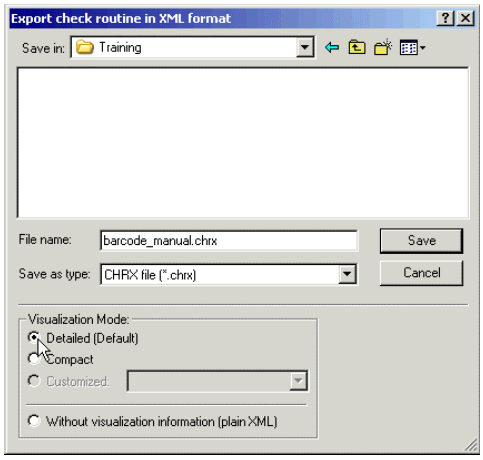
❻ The file just created can now be viewed using Internet Explorer. Double-click the created `.chrx` file in Windows Explorer. The Internet Explorer will start and display the exported check routine. Scrolling up and down, you can view the entire documentation and parameter settings for each check function. Furthermore, you can print the entire document from Internet Explorer using **File ▸ Print**.



❹

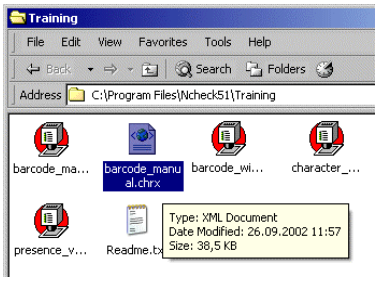
Using the **Documentation** tab in Structure View, check function descriptions can be viewed.

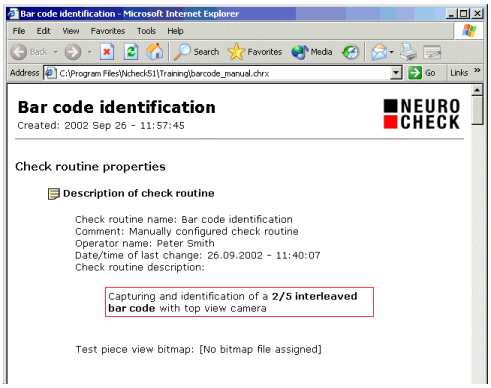




❺

Using the menu item **Check Routine ▸ Export ▸ as XML...**, a check routine can be exported for documentation purposes to XML format that can be read e.g. by the Internet Explorer. Choose **Detailed** for the **Visualization Mode**.





❻

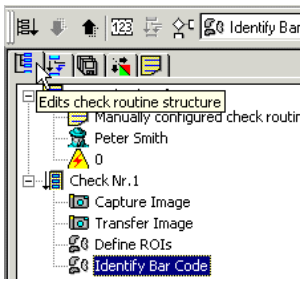
During NeuroCheck installation, the `chrx` files are associated with the Internet Explorer and can be opened by double clicking.

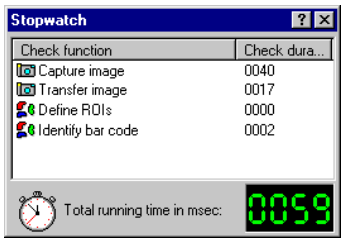
❼ When doing a feasibility study, it is not only important to give a description of the check routine but also to be able to roughly estimate the execution time of the inspection. For this, NeuroCheck provides a method for measuring time:

Choose **Tools ▸ Stop Watch** or the depicted icon from the tool bar to open the stop watch window. In the tree view, switch to the first tab **Structure**. Execute the individual check up to the last check function by selecting the check function Identify bar code and clicking the appropriate icon. In the stop watch window the execution time of each check function is displayed in milliseconds and also the total time of the individual check. This information can now be used in your documentation (e.g. in the HTML description of the individual check, see further down) to estimate how many test pieces can be checked per second or how much processing power will be required to satisfy the specifications of the application. Please note that the time for one individual check is measured, not the total cycle time. In automatic mode, the time for communicating with the PLC, for displaying output windows or for interruptions caused by the operating system must be added.

❽ Add the following text to the description of check function Identify bar code via the context menu: “This check function requires <u>approx. 2 msec for execution</u>.” The HTML tag “<u>” formats text passages as underlined. Activate the HTML check box and confirm with **OK**. Repeat the XML export as described in ❺ and confirm Overwrite with **Yes**.

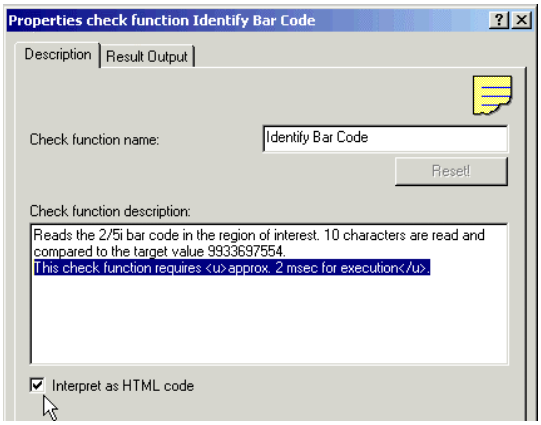
❾ Switch to the Internet Explorer and choose **View ▸ Refresh**. The changes to the description are displayed at the corresponding spots.





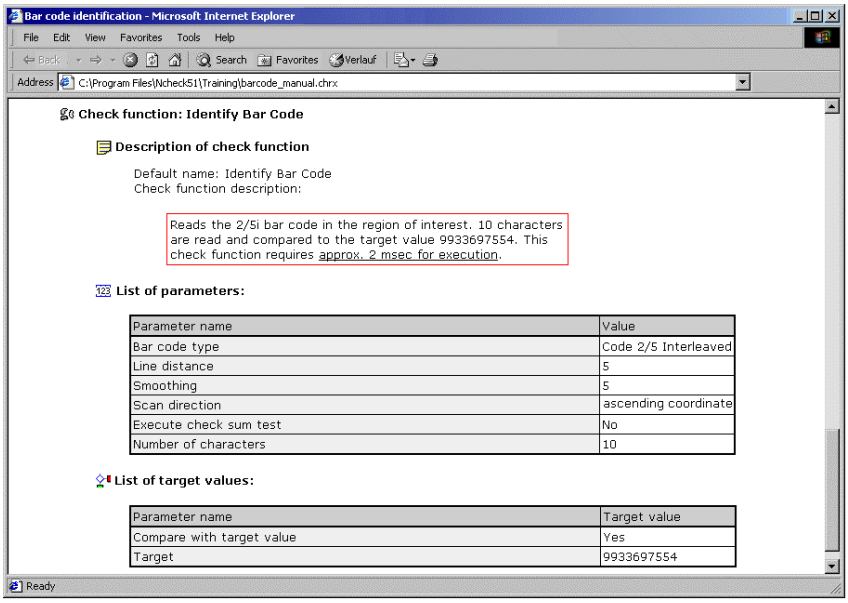
❼ 

Using the stop watch, you can determine the required inspection time. Execute the active check up to the last check function.



❽

Add the execution time to the description in your documentation and carry out another XML export.



❾ 

Refresh the view in the Internet Explorer. The exported documentation can be displayed and printed if need be.

3 Automatic Mode

The main application area of NeuroCheck is the automatic visual inspection in manufacturing processes. For a seamless integration into the production line, NeuroCheck is able to communicate with master computers or PLCs.

In this chapter you will learn

- how to load the device drivers necessary for communication,
- how to remote-control NeuroCheck from a PLC or master computer,
- how to automatically change check routines,
- how to configure output windows for automatic mode,
- how to assign commands to function keys.

Industrial PC with NeuroCheck running in automatic mode

3.1 Configuration of Digital I/O for Communication

Objective

You want to establish communication between NeuroCheck and the control units of your manufacturing process (PLC/master computer) via digital I/O.

Result

You can register, configure and test a digital I/O board in NeuroCheck. Furthermore, you will get to know NeuroCheck's virtual digital I/O that you can use for the training examples if you don't have the appropriate hardware installed in your system.

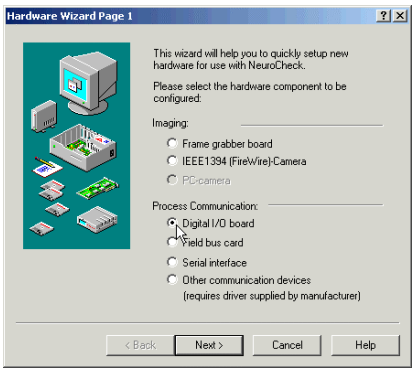
Solution

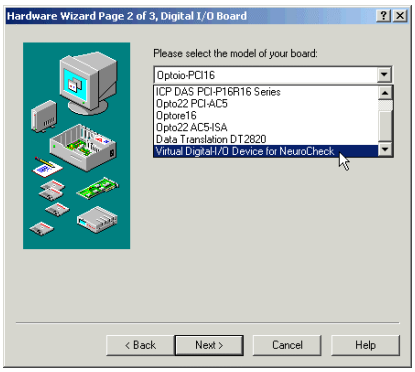
❶ From the **System** menu choose **Device Manager**. In the **Device Manager** dialog choose **New**. On the first page of the Hardware Wizard select **Digital I/O**. Choose **Next** to go to the next page. Here you can select the type of board you are using. For this example, select **Virtual Digital I/O**. Click **Next** and **Finish** to confirm.

❷ If you want the program simulating the opposite end of the digital I/O connection to be started simultaneously with NeuroCheck, confirm the following dialog with **Yes**. The simulation is opened and displayed. Select **View ▸ Always on Top** from the menu of the simulation program for the window to be permanently visible.
In NeuroCheck's **Device Manager**, select the entry for the digital I/O just created, and choose **Test**.

❸ In the window now displayed, you can set and reset the outputs. The remote station will display this change. You can set the inputs in the window of the remote station simulation. NeuroCheck displays the changes of the input signals in the test window.
If you have a real digital I/O and maybe a test switch board at your disposal, you can proceed in a similar manner. Thus you can easily check whether the communication between NeuroCheck and your PLC works properly.

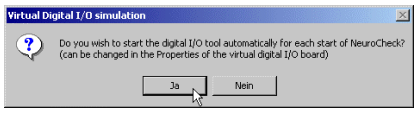
Close the dialog box and the device manager by clicking **OK**.

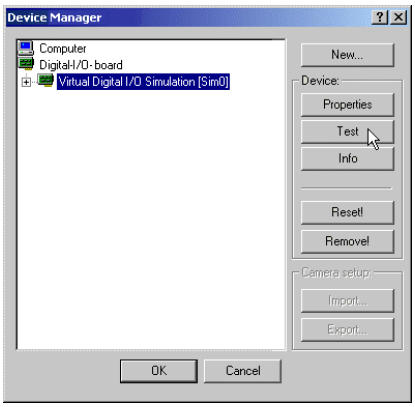




❶

Use the hardware wizard to register a digital I/O with the system. Here, NeuroCheck's virtual digital I/O is used.

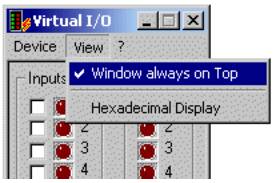


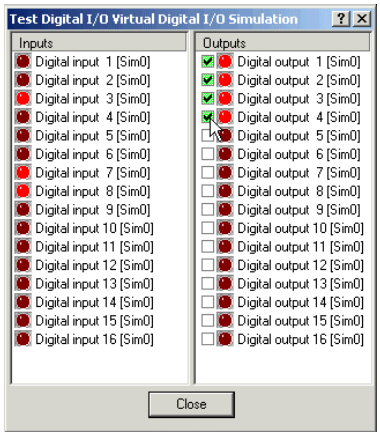


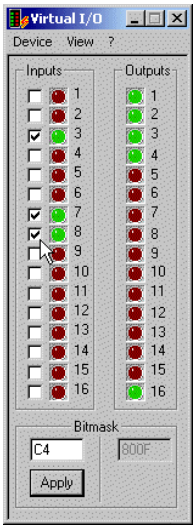
❷

After closing the wizard, the device is available in the hardware manager. The virtual digital I/O's remote station is displayed. Select: **View ▸ Always on Top**

Select the new entry and choose **Test** to test the operation of the device.







❸

Here the remote station simulation of the virtual digital I/O is used. The input signals to NeuroCheck can be changed, and the output signals from NeuroCheck can be displayed.

3.2 Remote Control

Objective

After creating the prerequisites for a communication with NeuroCheck via digital I/O in the previous chapter, you can now configure NeuroCheck in such a way that it can be remote controlled from the virtual remote station or a PLC.

Result

You can alter the settings for starting a check via digital I/O.

Solution

❶ From the **System** menu choose **Remote Control**. In the **Remote Control** dialog box switch to page **Input Signals** if it is not being displayed already. Activate the **Start check** signal by clicking the check box in front of the signal name. Select **Change** and choose **Digital I/O board** from the dialog box **Select Signal Source**. NeuroCheck then starts an inspection run if the start signal is set at the digital input. By default, the first input is the start signal. Choose **Options** to select a different input if necessary.

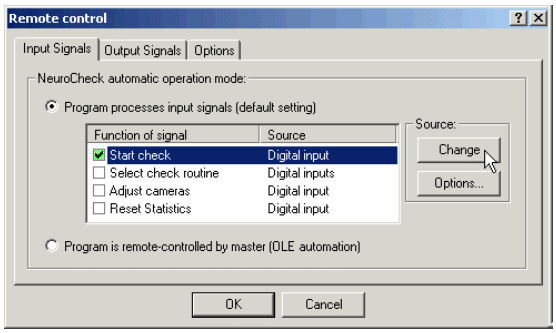
If you also activate the check box **Select check routine**, the check routine can be changed in automatic mode upon a signal from the digital I/O or serial interface. However, in that configuration the signal must be present. Therefore, we will leave this option deactivated for now.

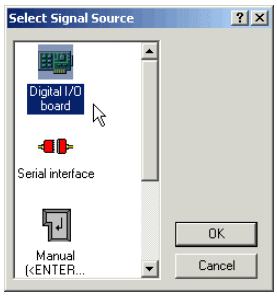
❷ Switch to the **Output signal** page and check **Check result OK/NOK**. This tells NeuroCheck to set the assigned output for ‘OK’ or ‘NOK’, signaling whether the check piece was all right or not. Furthermore, you can activate the output for **System ready**. NeuroCheck then sends a signal as soon as it is ready for a check, i.e. if NeuroCheck is in automatic mode and no check routine is running. You can understand this communication best by using the remote station simulation. The output signal **System running** is used to tell the PLC that NeuroCheck is present regardless of the current operating mode.
The default output target is the digital I/O. Else select **Change** and choose **Digital I/O board** from the dialog box **Select Signal Destination**.

❸ Using the **Options** button, the outputs to be used for the different output signals can be configured. The defaults are:
1 for ‘System ready’, 2. for ‘OK’, 3 for ‘NOK’, and 4 for ‘System running’.

Press **OK** to confirm your changes.

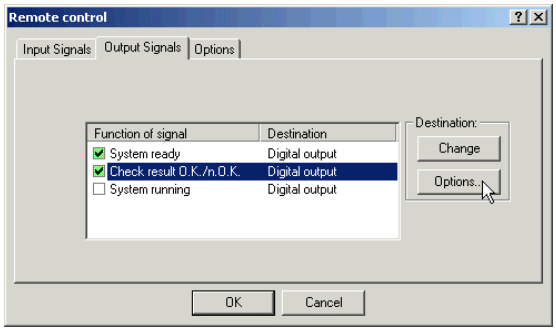
❹ Using the remote station simulation, you can now remote control NeuroCheck. This means: NeuroCheck executes inspection cycles in automatic mode as long as the check box for input 1 is activated.

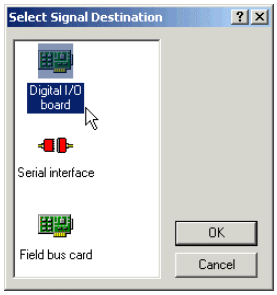




❶

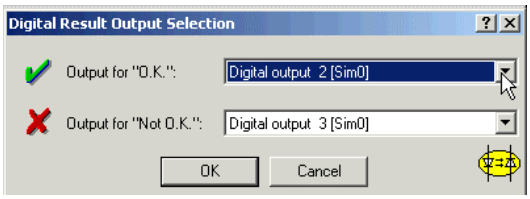
On page **Input signals** you can configure how NeuroCheck reacts to external signals.





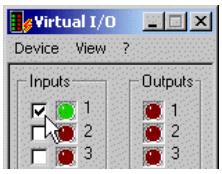
❷

On page **Output signals** you can determine what signals NeuroCheck will send during operation.



❸

Use **Options** to determine which output to use for which signal.



❹

Use the remote station to start inspection cycles in NeuroCheck.

3.3 Automatic Inspection Procedure

This chapter describes once more in detail the operation in automatic mode. Communication can be run via digital I/O, field bus or serial interface (cf. chapter 9.4). Here the virtual digital I/O and the remote station simulation are employed for process control and output signal display.

❶ At first NeuroCheck is in **Manual mode** or **Configure automatic screen mode**. Only the output signal 'System running' is present.

❷ 
After switching into **Automatic mode**, NeuroCheck sets the output for 'System ready' and waits for a start signal from the remote station. (In principle, timer-controlled and manual starts via keyboard are also possible, but not very useful in an automatic inspection application.)

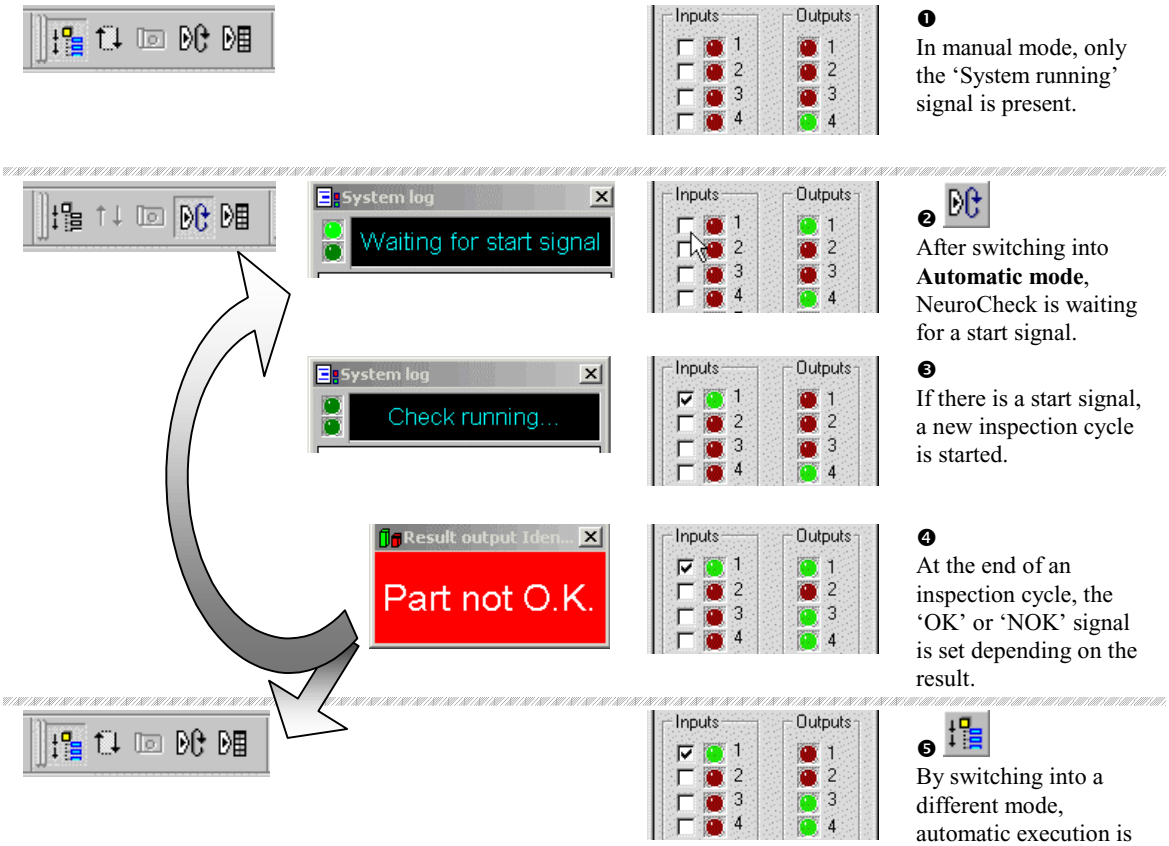
❸ If a signal is recognized, NeuroCheck deletes the outputs 'OK' and 'NOK' and the 'System ready' signal, and executes the check routine. NeuroCheck sends results to the various windows of the automatic screen, possibly writes result data to file, and determines the final result of the check. That means, as soon as an individual check yields 'NOK', the entire check cycle is determined 'NOK'.

❹ This result is sent to the destination set in the **Remote Control** dialog box. It can be identical to the start signal source though it does not have to be. In our example, either the output for 'OK' or for 'NOK' is set. After completing a cycle, the 'System ready' signal is set again.

Note:
Output of result data and check result to serial interface has been coordinated with check result output to digital I/O for high speed applications to prepare PLC or master computer for the input of result data via serial interface. If, in the **Remote Control** dialog box, digital I/O has been selected as destination for the check result signal, NeuroCheck will not send data via the serial interface while the check routine is running. Instead, it will first determine the final check result and send this via digital I/O, then the result data. Thus PLC/master computer can deal with other tasks during that time without risking to lose result data.

❺ 
If you switched to **Manual mode** or **Configure automatic screen** in the meantime, NeuroCheck stops the execution of the check cycles. Otherwise, processing is continued again at ❷.

For further information with regard to process behavior, see chapter "Automatic mode – Automatic inspection procedure" in the user manual or online help.



❶ In manual mode, only the ‘System running’ signal is present.

❷ After switching into **Automatic mode**, NeuroCheck is waiting for a start signal.

❸ If there is a start signal, a new inspection cycle is started.

❹ At the end of an inspection cycle, the ‘OK’ or ‘NOK’ signal is set depending on the result.

❺ By switching into a different mode, automatic execution is interrupted.

3.4 Automatic Check with Type Change

Objective

If you want to run different automatic checks on one system, e.g. if various types of one piece must be checked (e.g. cog wheels with 10 or 14 teeth), then NeuroCheck can perform a type change depending on the input signals. That means, during automatic mode a check routine is to be loaded and executed depending on a type number (ID) given by the PLC.

Result

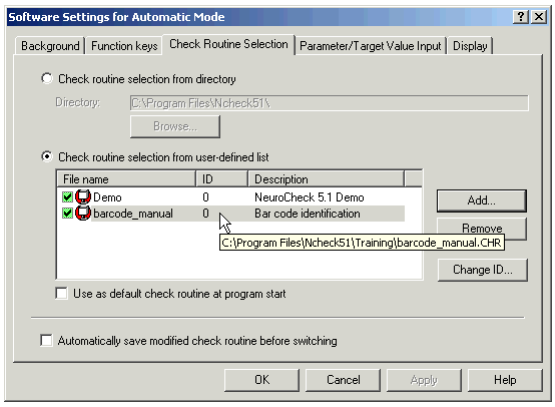
You can configure check routines and remote control for a type change.

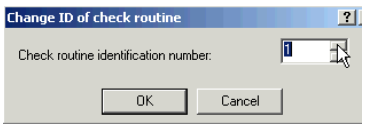
Solution

❶ First, the check routines concerned must be selected. Choose **Settings for Automatic mode** from the **System** menu. Switch to the **Check Routine Selection** tab. Activate the option **Check routine selection from user-defined list**. Press the **Add** button to open a file selection dialog with which to add one or more check routines to the list.
In this example, we first add the check routine `demo.chr` from the `Examples` directory and afterwards `barcode_manuell.chr` from the `Training` directory. Now, different IDs have to be assigned to all items in the list, otherwise the dialog cannot be confirmed with OK. For this, select **Demo** and click **Change ID.** Enter '1' as the new ID and confirm by pressing **OK**. Leave the settings dialog also with **OK**.

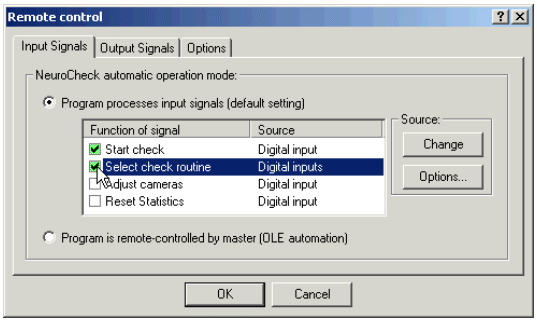
❷ Next, the remote control must be configured. In **System ▸ Remote control** switch to the **Input signals** page. Activate the check box **Check routine selection**. Use **Options** to see which inputs encode the type ID. 8 inputs, 3 to 10 are pre-selected. Thus 2^8=256 possible IDs can be binary encoded. The type ID can be between the values 0 (no input set) and 255 (all inputs set).

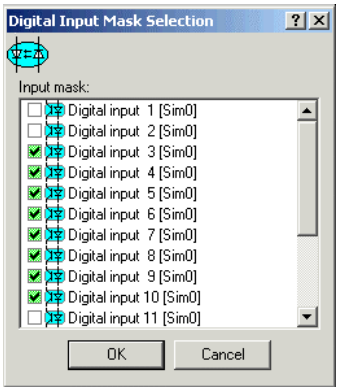
❸ Switch to **Automatic mode**. As soon as the start signal is present, the check routine selection signal containing the ID is evaluated. The ID is then looked up in the previously configured list, the corresponding routine is opened and executed. The remaining process is identical to the one described in the previous paragraph.
If you set in the remote station the input 3 active and inputs 4-10 inactive before setting input 1 for the start signal, type ID 1 is present. The demo check routine is opened and executed accordingly. During automatic mode, if you reset input 3 of the remote station , type ID 0 is present, and the bar code check is opened in the following inspection cycle.
If you choose any other signal combination for the inputs 3-10, a different type ID than 1 or 0 is present and the type change is unsuccessful since only the two check routines with ID 1 and 0 are configured in the list. This error is then noted in the system protocol.



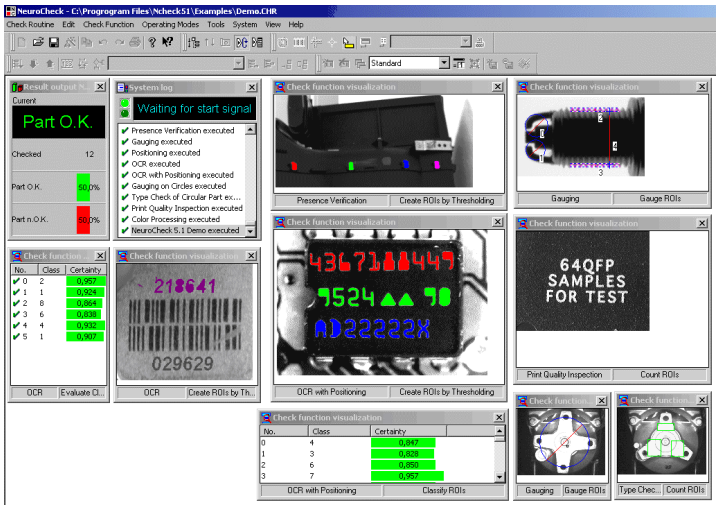


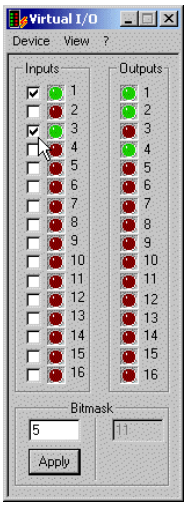
❶ Definition of possible check routines for type changes. Use **Change ID** to change the ID of the selected check routine.





❷ Settings for remote control for a type change using signals from a digital I/O. **Options** determines the inputs that encode the type ID.





❸ 

By switching input 3 of the remote station, the type ID is switched between 1 and 0. Depending on this, the demo or bar code check routine is executed.

3.5 Output Window Configuration

You have learned how to configure output windows in automatic mode in paragraph 2.3. Now we want to show you how to create an additional result window for the output of custom-made messages, and another check function visualization window for the graphic display of the bar code.
You can still use the check routine `Barcode_manual.chr` from chapter 2.4. Instead of capturing images from the camera, you can also use the bitmap `ncbarcod.bmp` from the `Training` directory.

❶ Switch to **Configure automatic screen** mode.

Choose **New ▸ Automatic Mode Output Window** from the **Edit** menu and select the **Result output** window from the **Add New Output Window** dialog box. The window is

created with a default size at a default position. You can also click the corresponding icon on the output window tool bar and open another window on the screen by dragging with the left mouse button. Note that the window has to be drawn with a certain minimum size.

❷ Click the window with the right mouse button and choose **Parameters** from the context menu. The Wizard for Result Output Windows is opened.
Choose the category **Status message** and click **Next**.

❸ As data basis for the output, you can use a check function, an individual check or the entire check routine. Select the check function Identify bar code and close the wizard with **Finish**. This window will now display text messages in automatic operation depending on the success or failure of the check function Identify bar code. Standard messages are used as default.

❹ To use custom-defined text for output, switch to manual mode.

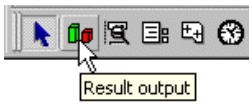
Select the check routine that we configured the output for Identify bar code and open the context menu by right-clicking. Choose Properties and switch to the **Result output** page of the dialog. There are three possibilities for the **Message text**:

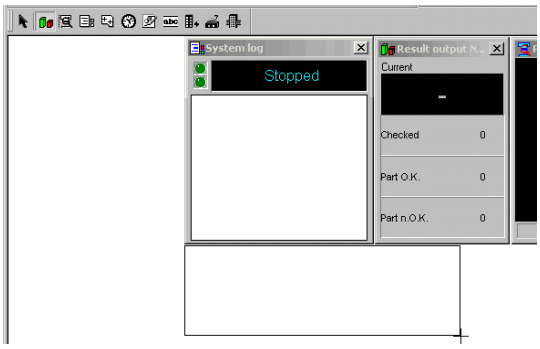
1.) "OK": the check function was successfully completed.

2.) “NOK”: during execution of the check function an error occurred. For example, the bar code does not correspond to the previously selected type.

3.) “Target value violation”: If the check function can complete its execution successfully but a previously set target value does not correspond to the resulting value, a target value violation is created.

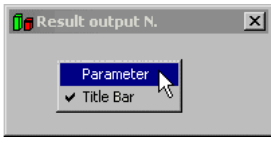
You can define a text for each of these possibilities that is then displayed in the status message window if the automatic mode window was configured appropriately.
Enter the following texts for our example: “Bar code read”, “Bar code cannot be identified”, and “Read wrong number”.

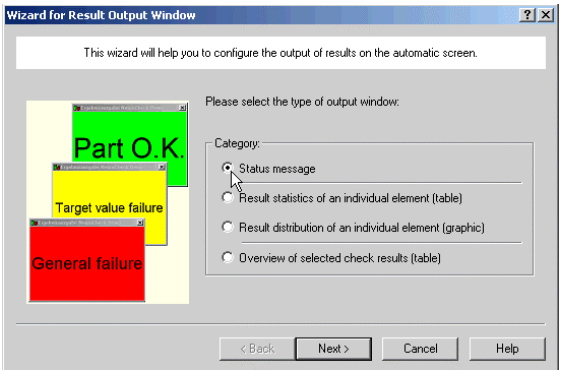




❶ 

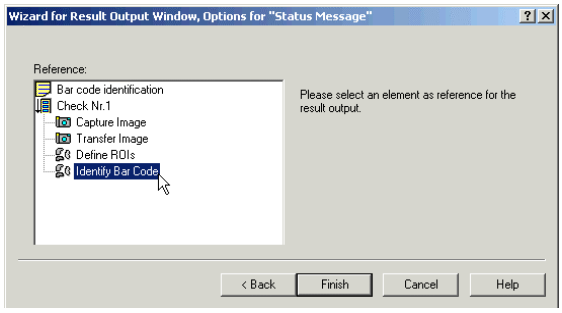
In **Configure automatic screen** mode you can click the appropriate icon to open a new result output window





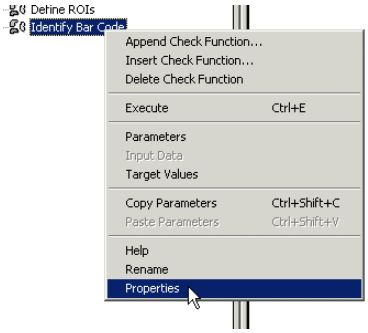
❷

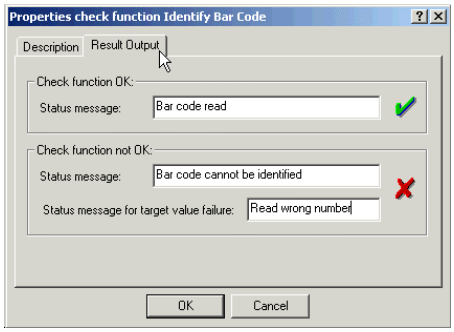
Use the **Parameters** command from the context menu to open the result output window wizard. Here you can configure the output.



❸

Select the check function Identify bar code as source for status messages.





❹ 

Define the text for a status message in manual mode. Enter message texts on the **Result output** page in the properties dialog of the check function Identify bar code.

❺ 

Next, we want to create a window for displaying the current camera image. We use the result output of the check function Define ROI as source since then the region of interest will be displayed that is the basis for the bar code identification.
Switch to **Configure automatic screen** mode. Click on the icon for a check function visualization window on the task bar and draw the window in the desired location.
Open the context menu within the created window and select **Parameters**.

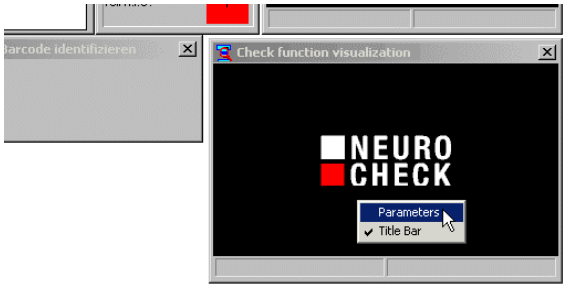
❻ In the tree view of the dialog box **CheckFunction Result Window Parameters**, activate the check box of the check function Define ROI. Leave the option for **Display** set to **Regions of interest** and the **Zoom** at **100%**.

Confirm with **OK**.

❼ 
Switch to **Automatic mode** and send a start signal using the **Enter** key or the start input of the digital I/O depending on your current configuration.

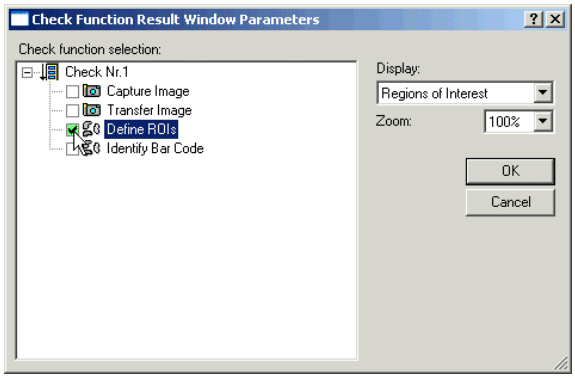
The visualization window now displays the result of the check function Identify bar code. Statistics are updated during every cycle and the status message window displays your custom-defined text depending on the check result.

Try to achieve different evaluations of the final result by using different bar codes as source images. When working with a camera, place different bar codes under the lens; when working with bitmaps, move the image section in the parameter dialog of the check routine Transfer image.
If you use the 2/5 interleaved bar code with the number “9933697554”, the message **Barcode read** is displayed; for the number “4000081258” **Read wrong number**; and for code 39 the message **Bar code cannot be identified**.



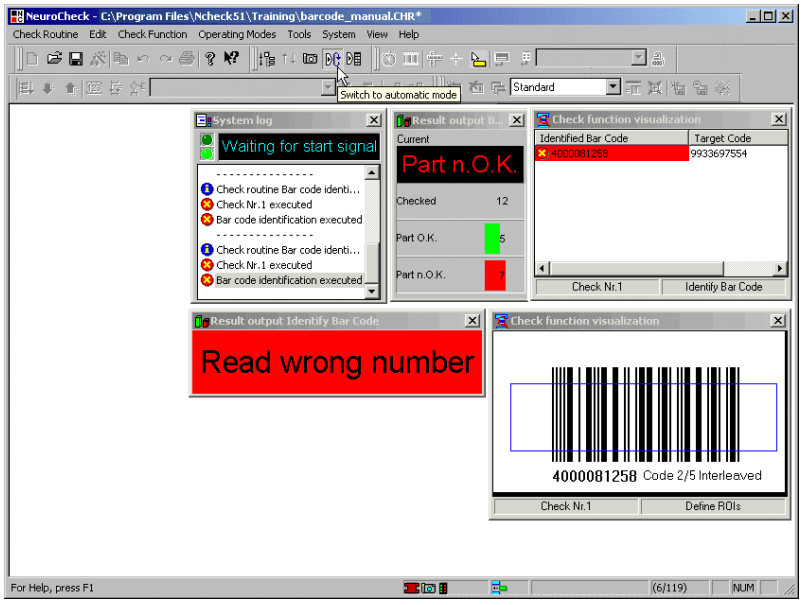
❺ 

Switch to **Configure automatic screen** mode. Activate the symbol for a check function visualization window on the task bar and draw the window at the desired location.



❻

Choose the check function Define ROI for visualization.



❼ 

Switch back to **Automatic mode** and send a start signal.

3.6 Function Key configuration

Objective

You want to set up the automatic mode in NeuroCheck as user-friendly as possible and control the system using easy keyboard commands. For example, you may want to start the NeuroCheck online help using the F1 key, reset statistics using F2, and finish automatic mode and switch into manual mode using F4.

Result

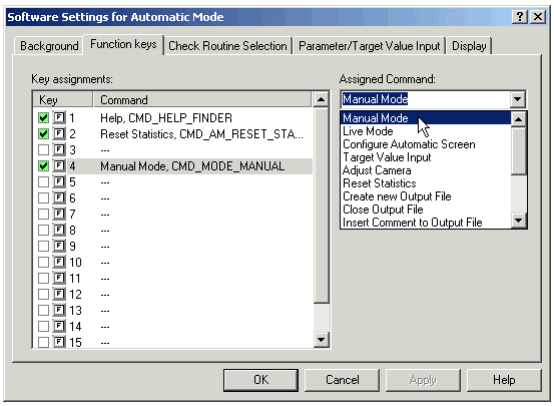
You can assign the function keys to various commands for automatic mode and create a standard HTML background.

Solution

❶ Choose **System ▸ Settings for automatic mode** and switch to the **Function keys** tab. In the key list select the first entry for the F1 key. In the combo box **Assigned command** you could, for example, choose the **Help** command. Select the second entry for the F2 key and choose the **Reset statistics** command from the list. For the F4 key choose the **Manual mode** command.

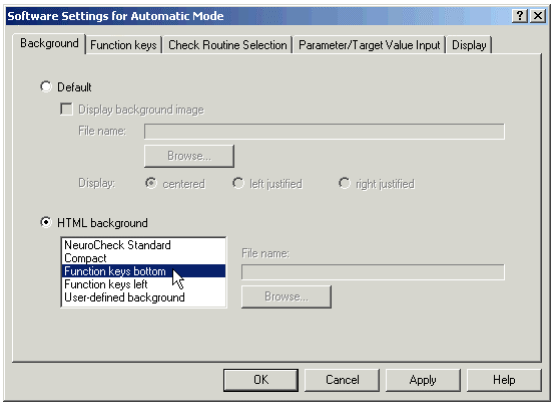
❷ NeuroCheck is now ready to be operated using the function keys. In addition, we want to use an HTML background for the automatic screen, on which the function keys and their commands are displayed.
Switch to the **View** tab and activate the option **HTML page**. Choose the background **Function keys bottom** from the list.
Confirm with **OK**.

❸ 
Switch to **Automatic mode**. You will see the function keys displayed graphically at the bottom of the screen. For each assigned function key, there is a text with the corresponding command. Since this is an HTML background, you can click on the texts and pictures to have the commands executed. In addition you can execute the commands directly via the function keys on your keyboard without using a mouse. For example, statistics are reset to 0 when you press F2.



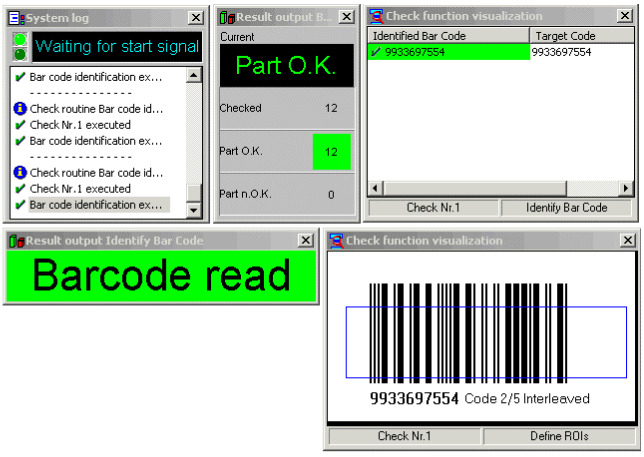
❶

Open via **System** ‣ **Settings for Automatic mode** ‣ **Function keys** the dialog for the configuration of the function keys. Configure the **F1**, **F2** and **F4** keys.



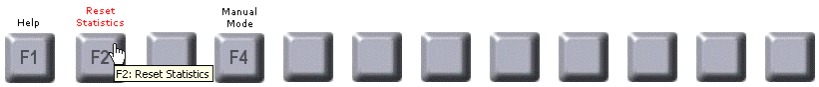
❷

You can configure the HTML background on the **Background** tab. Select the HTML page **Function keys bottom**.



❸

The elements of the HTML background are interactive, i.e., you can click on the pictures or the texts to execute the corresponding command. In addition, the commands can be executed directly by pressing the function key.



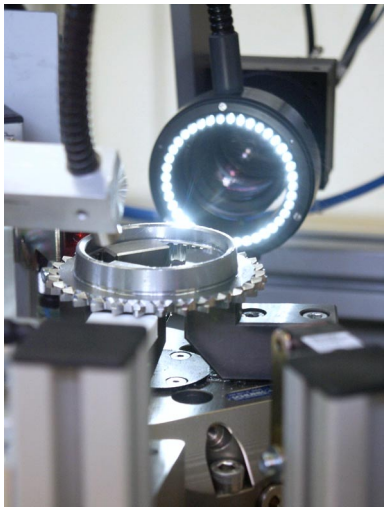
4 Presence Verification

In industrial vision applications a common task is to verify if certain parts of the object to be inspected are present. Usually first all objects are looked for that are worth considering. Then object features are used to distinguish between relevant and nonrelevant objects for counting.

In this chapter you will learn

- how to determine the presence of an object with NeuroCheck,
- how to configure a hierarchical object search, i.e. how to search for an object within another object,
- how to distinguish objects using feature values,
- how to select the individual checks to be executed in automatic mode in a check routine with several individual checks,
- how to have object counts displayed graphically on the automatic mode screen.

Most of the application samples in this and the following chapters use a 3.5" disk as test piece, because this object will most probably be available when working with a PC. On page 10-4 ff. you can find images of well-illuminated disks. The same images have been included as bitmap files with this training course, in case you do not have a camera available.



Visual inspection of a gear wheel

4.1 Determining the Presence of an Object

Problem

You want to ascertain that no disk leaves your house without being write-protected. This means that the write-protection slider of the disk has to be open.

Result

You can use NeuroCheck to detect the presence of an object and know how to tell NeuroCheck the number of objects that have to be present.

Solution

❶ 

Create a new check routine. Enter e.g. “Presence verification” as the name of the check routine and “Checks write-protection slider” as additional explanation. Change title and comment of the check for example to “Check write protection 1” (because we will do variations of it later) and “Uses fixed rectangle” respectively.

❷ 

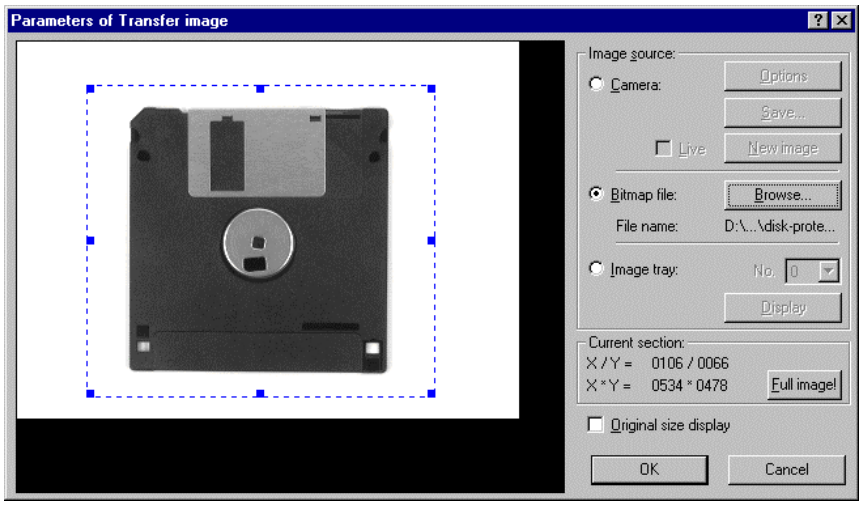

If you want to work directly from camera, you have to append the check function Capture image first; if not, leave this check function out. In either case the next check function is Transfer image. Depending on the configuration of your system, you can now capture a real disk with the camera or one of the disks shown on page 10-4 or use the bitmap file `disk-protected.bmp`. Adjust the image section so that the complete disk is being transferred.

❸ 


Append check function Define ROIs. Open the parameter dialog and define a rectangular region of interest enclosing the write protection opening. Define the rectangle so that it lies completely within the dark disk casing (the reason for this will become apparent later).

❶ 

Create a new check routine.



❷ 

Use check function Transfer image to load an image of a complete disk into memory.

❸ 

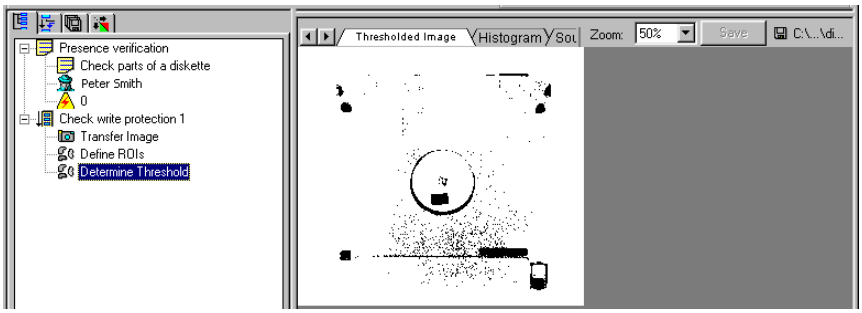
Define a rectangular region of interest enclosing the write-protection opening.

❹ Append check function Determine threshold. Again you will find this check function on the **Objects** page. This function analyses the brightness distribution in the defined regions of interest to determine a threshold for distinguishing objects and background. In this image, though, after executing the function you will notice that the determined threshold is not suited to separate the write-protection opening from the disk casing.

❺ The reason is easy to detect when you select **Histogram** from the tabs of the right window pane. You will see that the computed threshold is far too low because of the imbalance of the gray level distribution.

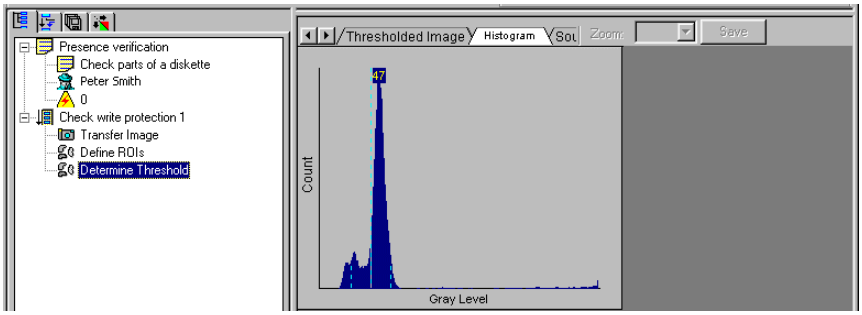
❻ The parameter dialog of function Determine threshold offers a solution for this problem. Choose the **Options** button in the **Automatic computation** section of the dialog. Choose **Strong** from the **Histogram equalization** list box. After leaving both dialogs and executing the function again you will notice that the imbalance in the histogram is much less pronounced.

❼ Switch back to **Thresholded Image** using the tabs of the right window pane to see the result. Obviously, the new threshold separates diskette and background much better.



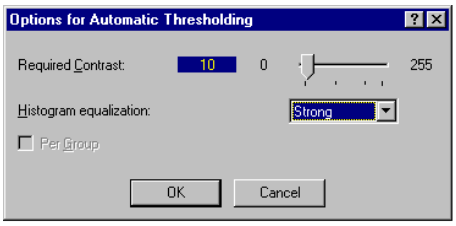
❹

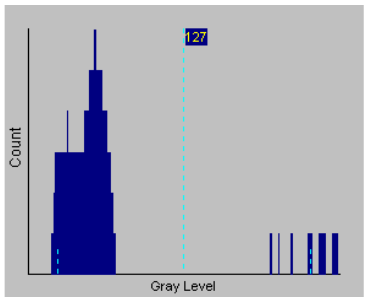
The default parameters of function Determine threshold yield no satisfactory result.



❺

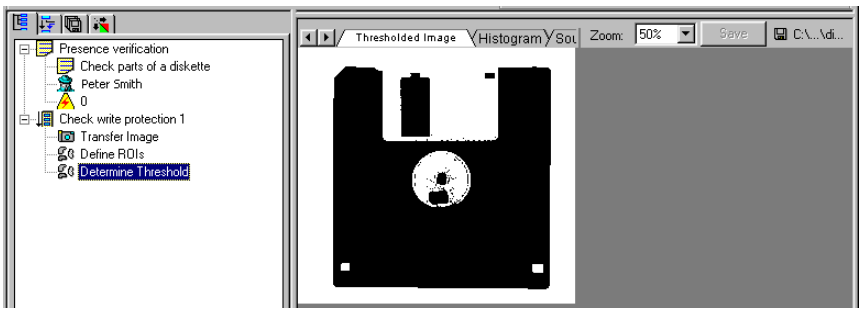
The histogram shows a strong dominance of dark pixels within the region of interest.





❻

Choosing **Strong** for the histogram equalization in the options dialog of function Determine threshold reduces the imbalance in the histogram.



❼

Segmentation by new threshold, computed with histogram equalization.

❽ 


Now append check function Create ROIs by thresholding from the **Objects** page and open its parameter dialog. Set the color of the objects to be found to **Light**, using the **Object color** list box or the context menu of the graphics panel. Leave the parameter dialog with **OK**. Execute the function. The opening appears in color to indicate that it has been recognized as an object.

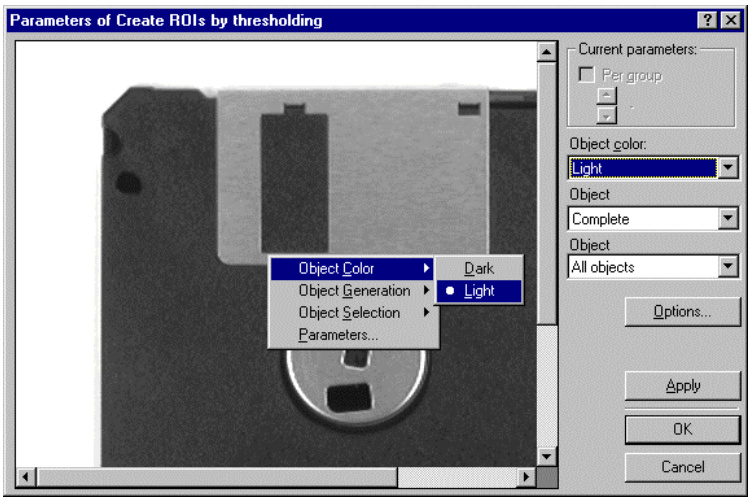
❾ 


Finally append function Count ROIs from the **Analysis** page to the check and open the parameter dialog. Activate the **Verify** button to instruct the function to check its result against target values. Otherwise the function would merely count the objects present in the image. Enter 1 for **Minimum** and for **Maximum**, because there has to be exactly one opening. Leave the dialog box with **OK**.

❿ 

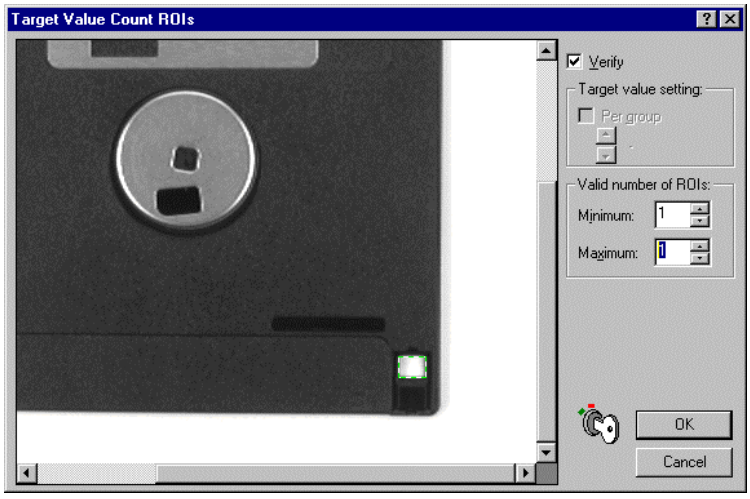
When you now execute the check routine up to this function and split the right window pane by dragging its double top frame down, NeuroCheck displays the input image with the object in the upper window region and a list consisting of one line in the lower window region. The list states that one region has been found and indicates the compliance of this object count with the prescribed values by a check mark and a green background. The write-protection opening in the opening also bears a green frame.

Now put a disk with a closed write-protection slider below the camera (or select in function Transfer image the bitmap file `disk-unprotected.bmp`). Make sure to position it precisely to keep the write-protection slider within the defined region of interest. Execute the check routine down to the final check function, Count ROIs. NeuroCheck reports an error because it could not find a write-protection opening.



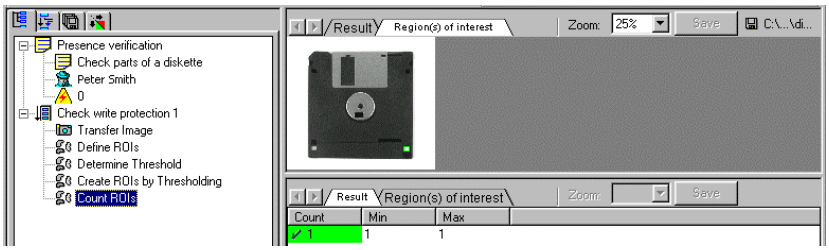
❽ 

Choose **Light** as the object color in the context menu or the list box of Create ROIs by thresholding.



❾ 

Configure function Count ROIs to verify the number of objects and set the allowed minimum and maximum numbers to 1 each.



❿ 

If the write-protection slider is open, NeuroCheck finds the opening as a light object and reports the correct number.

4.2 Presence Verification Using Object Properties

Problem

If the positioning of the disks under the camera cannot be guaranteed with sufficient precision, the write-protection opening does not always fit into the defined region of interest.

Result

You know how to configure a hierarchical object search in NeuroCheck, measure object properties and evaluate objects according to these properties.

Solution

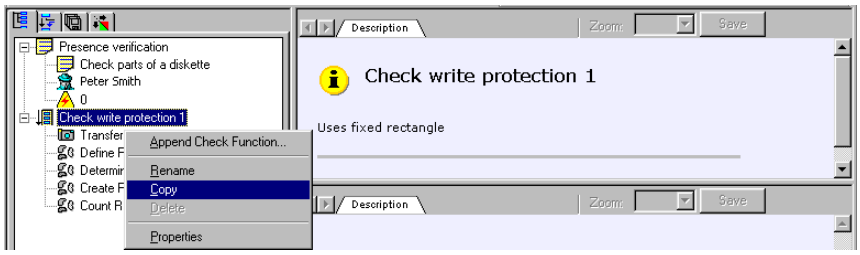
❶ 

At first the procedure is equivalent to the former. The easiest way, therefore, is to copy the check configured in the previous section by choosing **Copy Check** from the **Edit** menu or the context menu of the check in the edit pane or by clicking the depicted symbol from the edit bar. Rename the check and change its explanatory text, for example to “Check write protection 2” and “Uses hierarchical search” respectively.

❷ Select the last check function of the new check, Count ROIs, with the left mouse button and remove it by choosing **Delete** from the **Edit** menu or the context menu of the function.

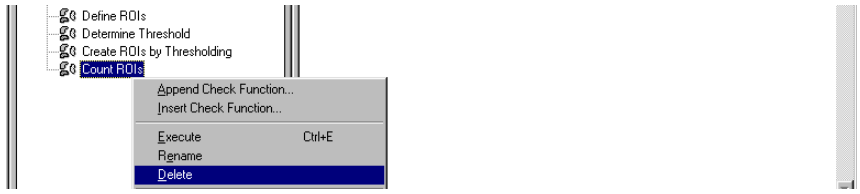
❸ 


Open the parameter dialog of function Define ROIs and choose **Select** mode. Then resize the rectangle around the write-protection slider to enclose the complete disk. The easiest way to do this is choosing **Full Image** from the context menu in the graphics panel. Confirm with **OK**.



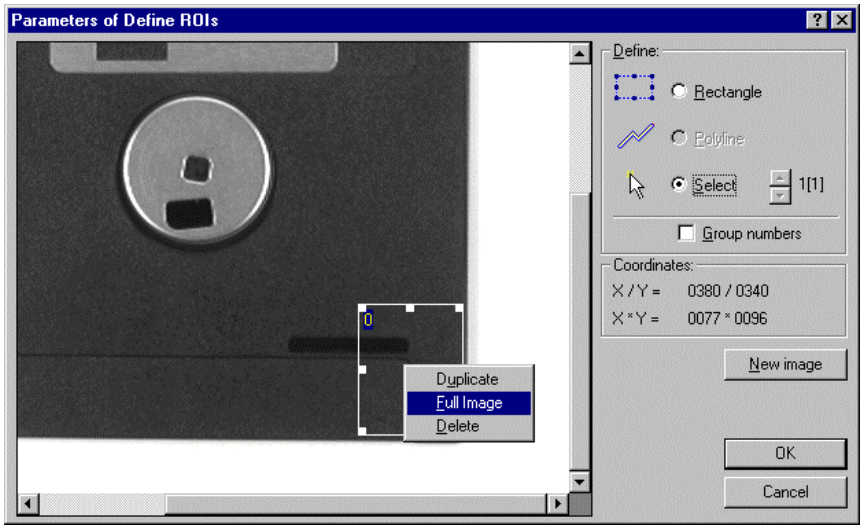
❶

Copy the finished check using **Copy Check** from the context menu or the **Edit** menu.



❷

Delete the last check function.



❸

Resize the rectangle to enclose the complete disk. The easiest way to do this is using the context menu of the graphics panel.

❹  

Open the parameter dialog of function Create ROIs by Thresholding. Change the object color to **Dark** and the object selection to **Largest object**. Executing the function now recognizes the complete disk as an object (the **Largest object** setting avoids artifacts due to illumination problems).

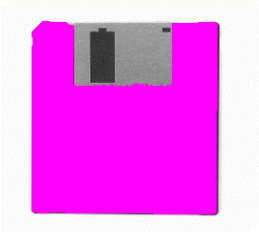
❺  

Now append function Create ROIs by Thresholding once more. Set the object color to **Light**. The function will find all white objects inside the dark object found in the previous step. You have just used one of NeuroCheck's most powerful features, the hierarchical object search. It allows you to search objects for details from the outside inwards (the circular drive plate in the center of the disk appears somewhat frayed. This is due to a not optimal setting of the binary threshold and will have no negative effect on the following steps).

What have we achieved? The check has a been changed in such a way that it finds the write-protection opening reliably, as long as the disk remains within the camera's field of view, whereas the first version of the check depended on very precise positioning. On the other hand, NeuroCheck now finds all bright objects within the limits of the disk, i.e. the write-protection opening, the other opening indicating high-density disks and the metallic drive plate in the center. In addition, some reflexes may appear as small objects depending on the illumination conditions. We thus need to find a way to distinguish the write-protection opening from other objects. It will be very difficult to distinguish it from the high-density marking, both openings being exactly identical in size and shape but this poses no problem, because the high-density-opening will always be present. It is nevertheless very important to remove the small artifacts for a reliable object count. Obviously the objects can be distinguished very easily according to their area.

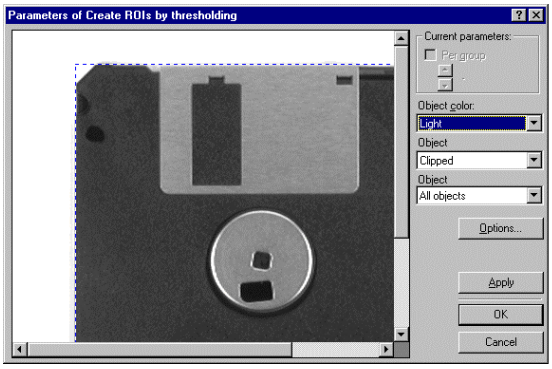
❻  

Append function Measure ROIs from the **Analysis** page and open the parameter dialog. Deactivate the preselected values **Origin X** and **Origin Y** by clicking the check box in front of the measurement description. Instead select the check boxes in front of **Area** and **Form factor** and leave the dialog with **OK**. If you do not find a measurement immediately, you can have the measurement list sorted according to the contents of a column by clicking the column title.



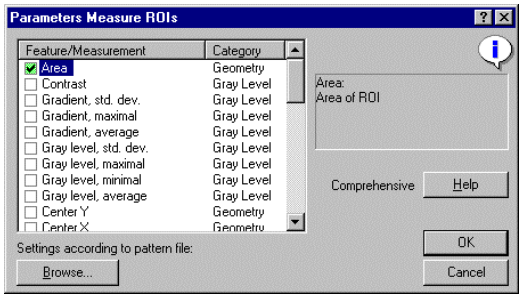
❹ 

Then change the object color to **Dark** to find the complete disk as an object.



❺ 

Append function Create ROIs by thresholding again for light objects. NeuroCheck finds every light object within the disk casing.



❻ 

Use function Compute Features to determine the areas and form factors of all objects from the previous step.

❼ 

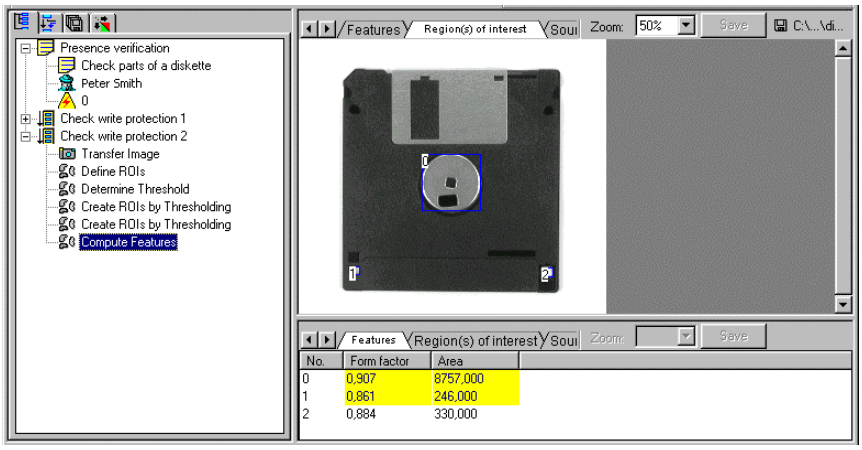
Executing the function creates a list with the areas and form factors of all the bright objects created in the previous step. The values characterize three very different types of objects:

- The drive plate with a large area and a very low form factor (due to the ragged edges created by the not optimal threshold value).
- The high-density and the write protection openings with medium areas and high form factors, indicating their clear geometrical shape.
- Some artifacts with small areas and medium to low form factors (because they are never quite as regular as the openings).

The highest and lowest values in each feature column are marked by a gray background for easy reference.

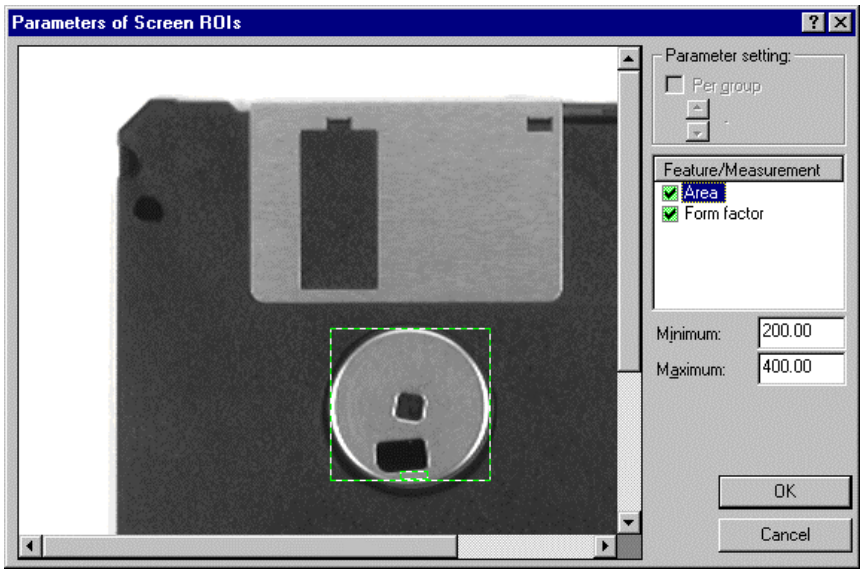
❽ 

Append function Compute Features and open its parameter dialog. Activate the check box in front of **Area** and set the minimum value to 200, the maximum value to 400. Do the same for **Form factor** with thresholds of 0.8 and 1.0 respectively. Executing the function removes all objects except the two openings because they do not comply to the screening criteria.



❼

NeuroCheck displays the existing objects and the pertaining measurement values in a numbered list.



❽

Screening with the corresponding limit values removes all objects except the two square openings.

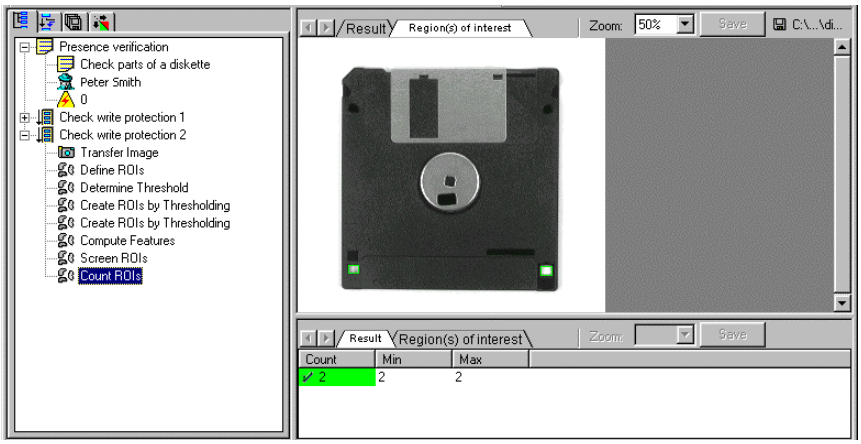
❾ 

As in the first version of this check the objects have to be counted, but now two objects are required because the high-density marking will always be present. Append function Count ROIs and enter 2 as the **Minimum** and **Maximum** object count.

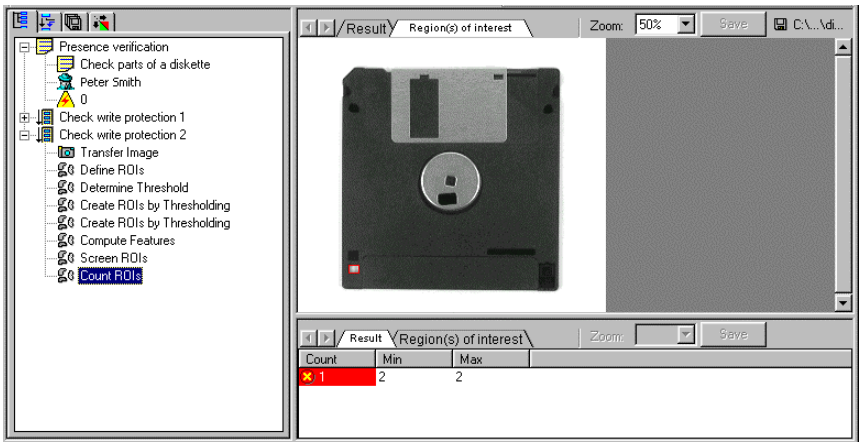
After executing the function NeuroCheck reports the disk to be O.K. since the two required objects have been found.

❿ 

Closing the write protection slider or selecting bitmap file `disk-unprotected.bmp` in function Transfer image causes an error to be reported, because only the high-density marking is present.



❾ The write-protection opening as well as the high-density marking comply to the screening criteria. With two objects the disk is considered O.K.



❿ Here the write-protection slider has been closed leaving only the high-density marking as an object. The disk is not O.K.

4.3 Inspection Statistics

Problem

You want the operator to be able to judge the performance of your production line at a glance.

Result

You are able to create new automatic mode screen layouts, configure textual and graphical output of inspection statistics to the automatic mode screen and know how to deactivate complete checks for automatic execution.

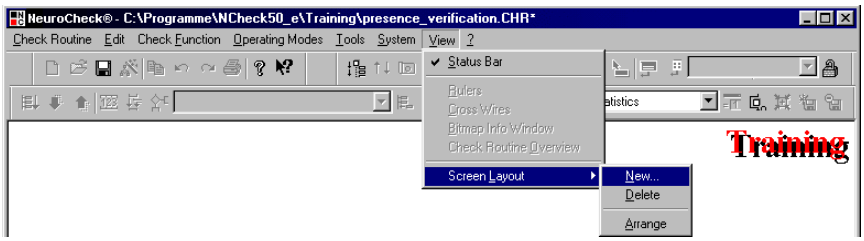
Solution

❶ Every check routine can manage arbitrarily many different screen layouts for different purposes. To create a new one choose **Configure Automatic Screen** from the **Operating Modes** menu. Then choose **Screen Layout ▸ New** from the **View** menu or click the corresponding icon.

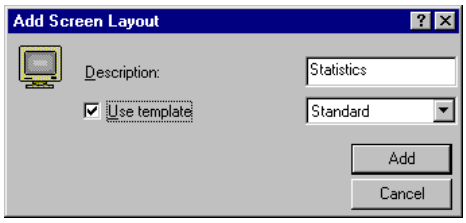
❷ In the **Add Screen Layout** dialog box, enter “Statistics” as description, activate the check box **Use template** and select the **Standard** template from the list box. Confirm with **OK**. NeuroCheck displays a new screen based on the standard layout.

❸ Open the context menu of the **Check function visualization** window and activate the checkbox of the check function Count ROIs of the second single check. Choose **Region(s) of interest** from the Display combo box.

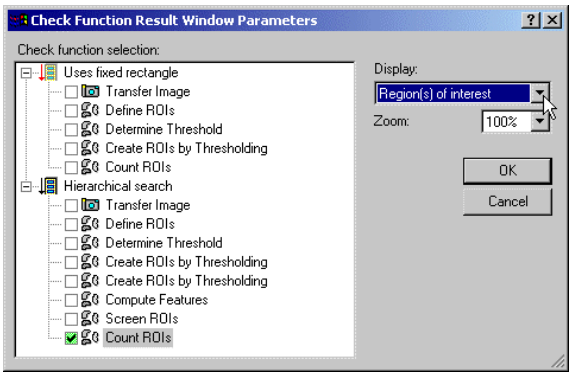
❹ Choose **New ▸ Automatic Mode Output Window** from the **Edit** menu and select the **Result output** window from the **Add New Output Window** dialog box. The window is created with a default size at a default position. Alternatively you can click the corresponding icon in the automatic screen configuration toolbar and draw the window at the appropriate location. Note that the window has to be drawn with a certain minimum size.



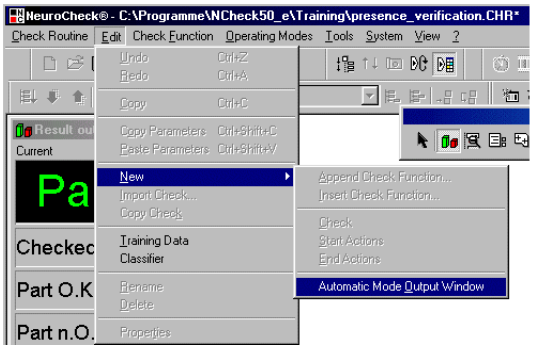
❶ Create a new screen layout



❷ Use the standard template as the basis for the new screen layout.



❸ Configure the Check function visualization window for the graphical output of the check function Count ROIs of the second single check.



❹ Add a new **Result output** window to the screen layout.

❺ Open the context menu of the new **Result output** window and choose **Parameters.** The Wizard for result output windows will be opened. Select the category **Result distribution of an individual element (graphic)** and choose **Next**. Select the first element of the list to display the results of the complete inspection run. Choose **Next** and **Finish**. This window will now display the percentage of the O.K. parts of all inspected parts as a bar chart.

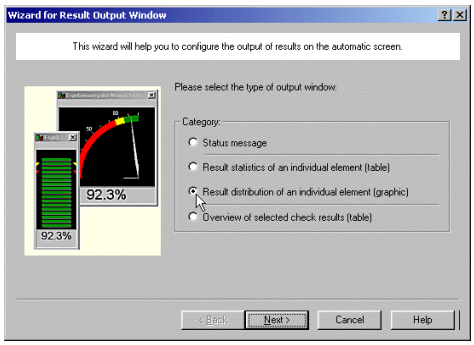
❻ 

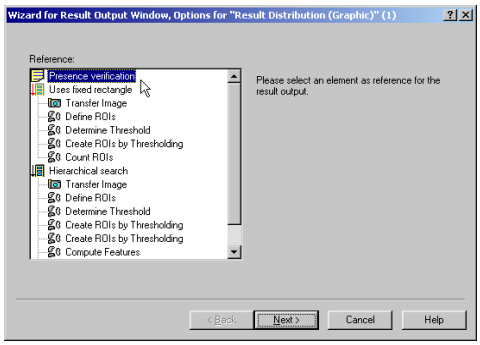
Not having to position a test piece precisely is of course of special importance in automatic mode, therefore only the second version of the check should be used. The first check should not be performed at all. Instead of deleting the first check, you can simply switch it off for automatic operation. This feature can be very useful if you need a certain image processing procedure only sometimes for adjusting the inspection system but do not want to reconfigure it every time.

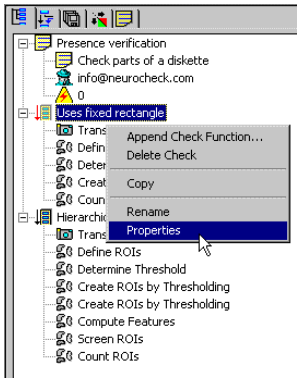
Switch back to manual mode and select the first check in the edit pane. From the **Edit** menu or the context menu of the check choose **Properties**. Switch to page **Flow control** and click the **Deactivate check** box. Now the check will simply be skipped in automatic mode. In the edit pane this will be indicated by a red arrow in the check icon instead of a black one.

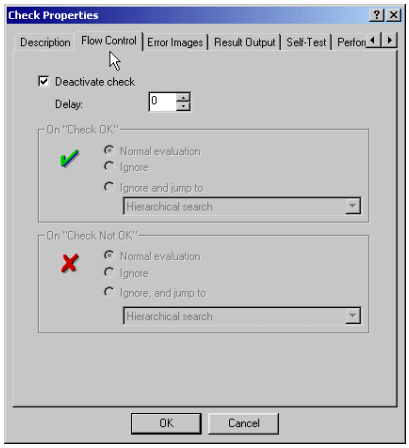
❼ 

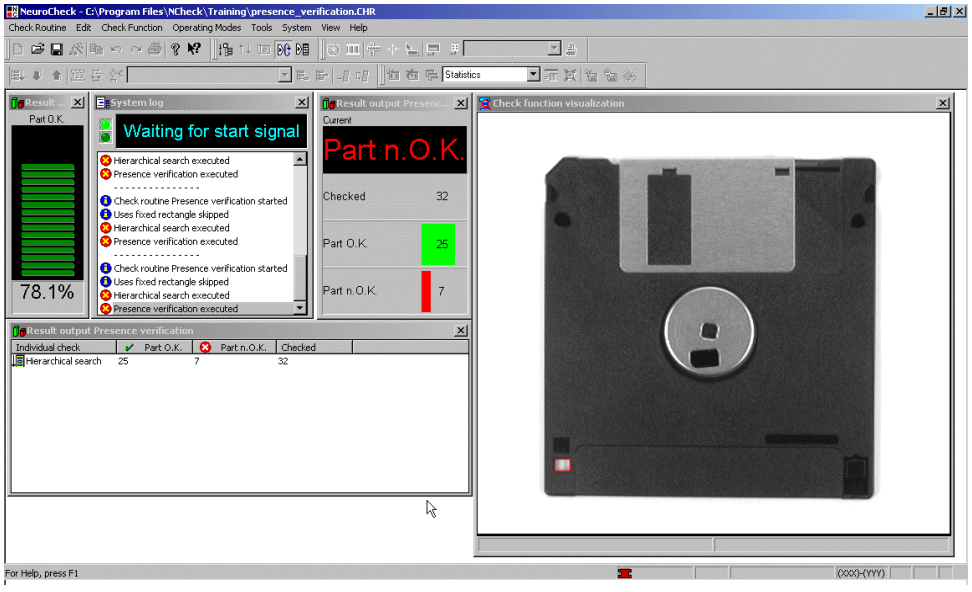
Switch to automatic configuration mode and arrange the windows in a convenient way. Now switch to automatic mode and execute the check routine a few times. To get different results you can switch diskettes in front of the camera or go back into manual mode and select different bitmaps in function Transfer image.











❺ Configure the new **Result output** window to display a bar chart presenting the percentage of inspected O.K. parts.

❻ Deactivate execution of the first check.

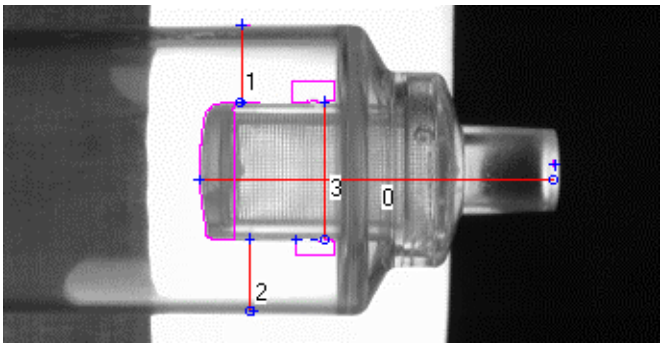
❼  View of the automatic mode screen after some inspection runs with varying check results.

5 Gauging

One of the most important applications of digital image processing is the contact-free gauging of work pieces. NeuroCheck uses special interpolation algorithms to determine object dimensions with a precision higher than one image pixel.

In this chapter you will learn

- how to determine distances between objects,
- how to create objects from edges, model these edges by straight lines and determine their orientation,
- how to write measurements to a file, e.g. for documentation and statistical evaluation in spreadsheet programs,
- how to change nominal values and allowances in automatic mode.



Gauging of a drip chamber

5.1 Measuring Object Distances

Problem

You want to ascertain that the drive plate in the middle of the disk has not been dislocated and moved under the edge of the opening in the disk casing. Geometrically this means:

1. When a part of the drive plate slides under the edge of the opening , its minimum dimension will be too small.
2. The distance between the drive plate and the lower edge of the slider has to be within a limited range.
3. Distances between the drive plate and the left and right edges of the disk have to be within a limited range.

Result

You are able to determine geometrical properties of a single object and distances between two objects with NeuroCheck.

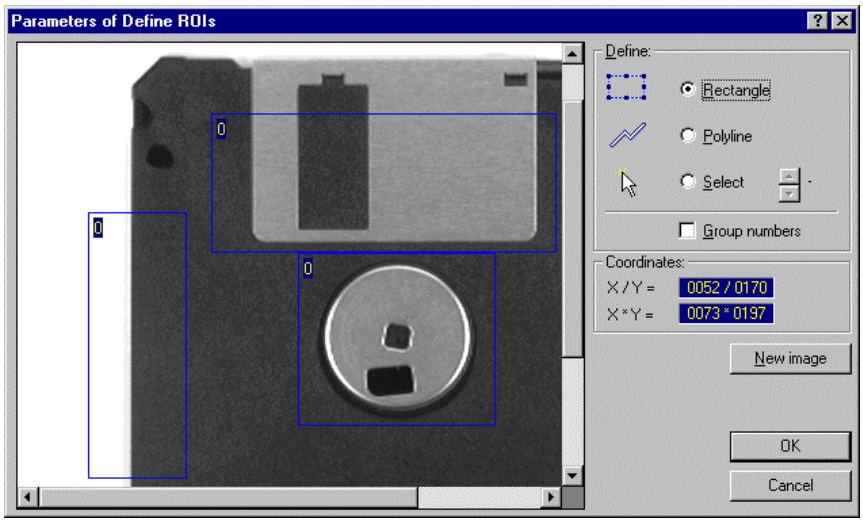
Solution

❶  

First proceed as in the previous chapter, i.e. transfer an image of a disk into memory, either directly from camera or from the bitmap file `disk-protected.bmp`, and append check function Define ROIs to the check. In the parameter dialog of this check function define three rectangular regions, one enclosing the drive plate, another crossing the slider, and a third crossing the left edge of the disk. Take care that the lower edge of the slider is completely inside the region. This will be very important for the gauging later on.

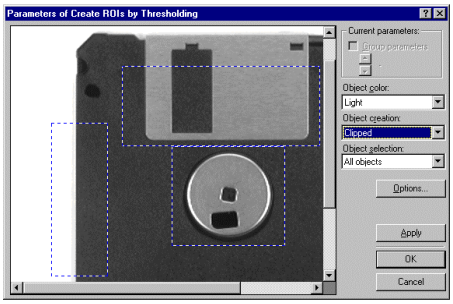
❷   

Append check functions Determine threshold and Create ROIs by thresholding. In the parameter dialog of the latter set the object color to **Light** and object creation to **Clipped**. It is very difficult to separate the slider from the bright background surrounding the disk. The **Clipped** setting allows NeuroCheck to cut detected objects at the limits of the regions of interest. Otherwise it would try to create a large white object surrounding the disk completely; such an object would be considered as background (and hence invalid), because it touches the full length of each image border. Executing the function now creates part of the slider and the background along the disk edge and the complete drive plate as separate objects.



❶

Define three rectangular regions of interest enclosing the drive plate, the left edge of the disk, and crossing the slider.



❷

Set the object color to **Light** and object generation to **Clipped** to prevent the slider from merging with the background and to cut out part of the background along the disk edge.

❸  

Now append check function Gauge ROIs. You will find this check function on page **Gauging**. The parameter dialog is mostly empty because no measurement has yet been defined. Choose the **New** button. The first page of the **Gauging Wizard** appears, where you can select the objects to be measured. First we want to determine the size of the (visible portion of the) drive plate. As a separate image object the drive plate is enclosed by a blue rectangle. Click this rectangle with the left mouse button. It will then appear dotted indicating that it has been selected. The topmost combo box in the **Geometrical descriptions** area becomes active. From this list select item **Contour** because we want to gauge the radius of the drive plate, which is a property of its boundary. Choose **Next**. On the second page of the Gauging Wizard select **Radius, minimal** from the **Gauging rule** box and enter a descriptive text in the **Text** box. Then choose **Finish**. The newly defined measurement will now appear in the list area of the parameter dialog with its identification number, the descriptive text entered on the final wizard page and the gauging rule used.

❹ 

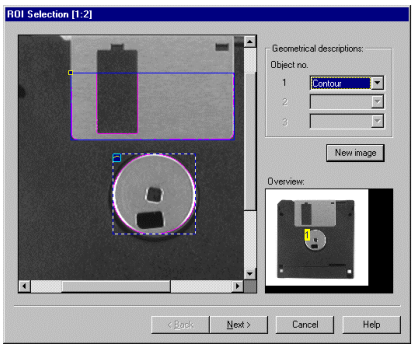
Choose **New** again. This time select the drive plate and the slider on the first page of the **Gauging Wizard**. For both objects select geometrical description **Contour**. Note the small blue/yellow icon in the top left corner of the objects indicating the geometrical description used. Choose **Next** to switch to the second wizard page. The correct gauging rule is already preselected: **Distance, minimal**, so you only have to enter a description for this measurement. NeuroCheck will now measure the minimum distance between the contours of the two objects, i.e. the minimum width of the gap between the bottom edge of the slider and the top edge of the drive plate. Choose **Finish** to close the wizard.

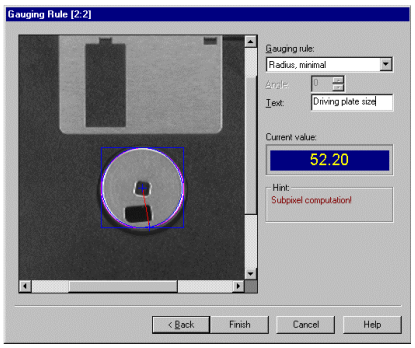
❺ 

Proceed as before, but this time select the object on the left edge of the disk and the drive plate. Set the contour as the geometrical description for both objects. Again **Distance, minimal** will be preselected on the second wizard page, so you will only have to enter a new text description for the measurement.

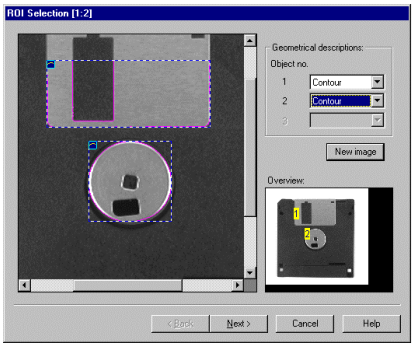
❻ 

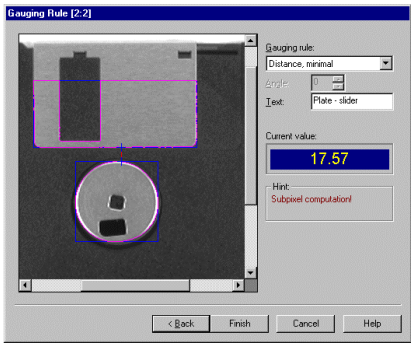
The parameter dialog now displays all three measurements with their identification numbers, text descriptions and gauging rules.



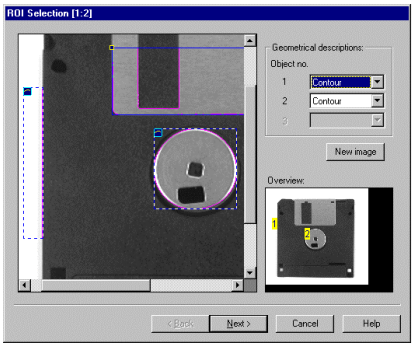


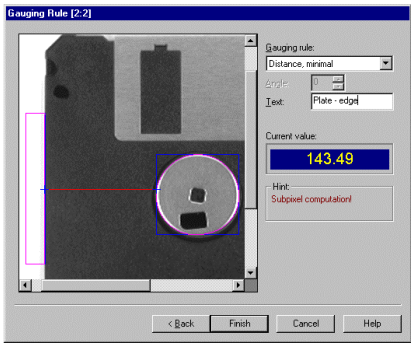
❸ Select the drive plate contour for gauging and choose gauging rule **Radius, minimal**.



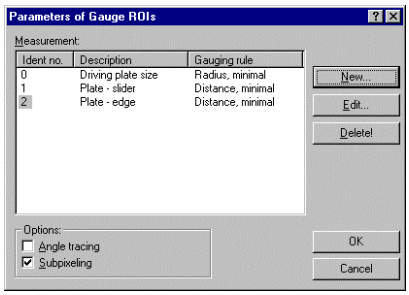


❹ Select the drive plate and the drive plate for gauging and confirm the preselected rule **Distance, minimal**.





❺ Select the object on the disk edge and the slider for gauging and confirm the preselected rule **Distance, minimal**.



❻ The parameter dialog now displays all three measurements with their identification numbers, text descriptions and gauging rules.

❼ After executing the function and splitting the right window pane in the usual way, you will see a list of the measurement values together with the description texts in one of the window regions, a graphical representation of the objects and measurements in the other one. Depending on your screen resolution you may have to adjust the zoom factor for the measurement graphics display.

❽ The main purpose of a gauging application is to compare the measurements with certain nominal values. In order to do this, append function Check allowances from page **Gauging** and open its parameter dialog. The dialog box displays a list of the available measurements with their description texts, current values, nominal values and allowances. The check box in front of the identification numbers of the measurements indicates whether a measurement is used for evaluating the test piece. Activate all three measurements by clicking the check boxes. Select the first measurement by clicking its identification number. Now you can enter the nominal value and the allowances.

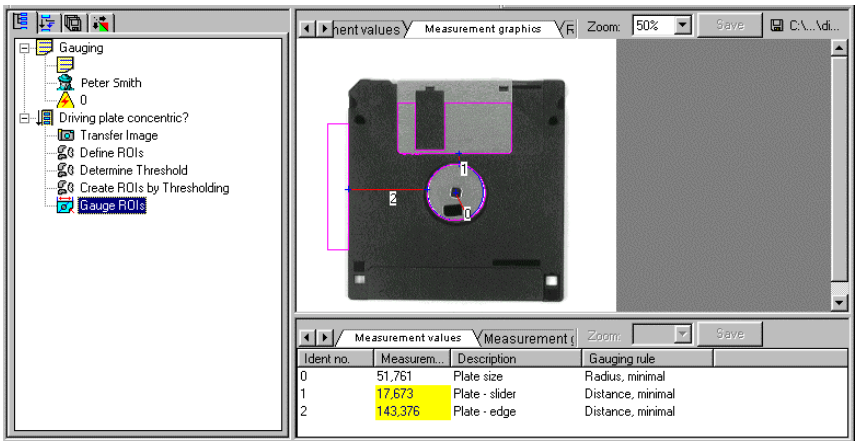
In a real application you would have to carry out a sequence of test runs with correct and incorrect disks to determine realistic values for the allowances – or take the values from a drawing, e.g. for a punched or die-cast work piece. In our example simply activate the **Percentage** button and enter the value 2 in the **Upper allowance** box and -2 in the **Lower allowance** box. NeuroCheck already entered the current value in the **Nominal value** box. You may want to adjust this value a bit, perhaps by removing the decimals.

Now select the second and third measurement in turn by clicking their identification numbers and proceed as before.

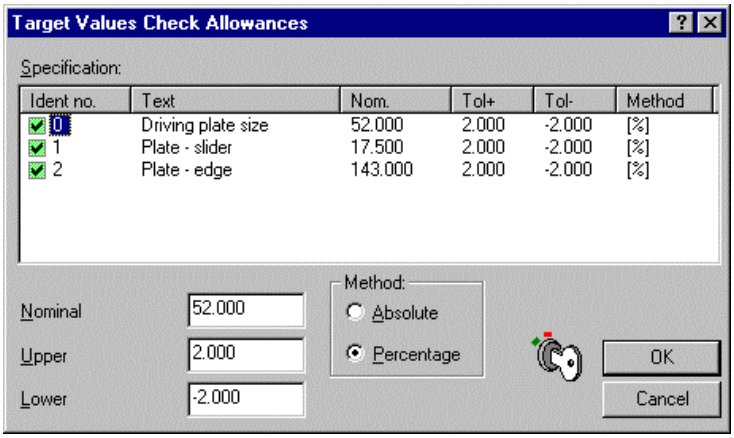
❾ Leave the parameter dialog with **OK** and execute the function. NeuroCheck displays a list of the measured values together with their description texts, nominal values and allowances. All values bear a green check mark because they do comply with the specifications. Try out the check routine with images of different disks or alter the nominal values and allowances and observe the behavior of the check routine.

Note that this function does not offer a graphical display of the measurements. The reason is that the measurements might be computed from other measurements using function Derive Measurements so that they do not need to have a direct correspondence to the image.

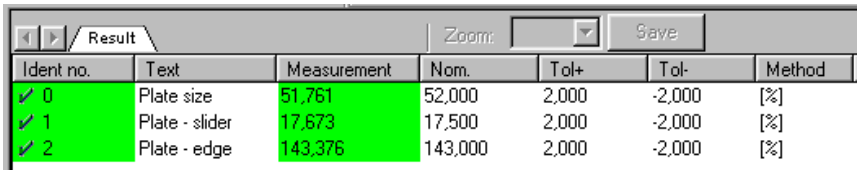
❿ Go back to function Transfer Image, choose **Browse** and select image `disk-drive-dislocated.bmp`. When executing function Create ROIs by Thresholding you will notice some small artifacts on the exposed disk surface inside the driver opening. These would confuse function Gauge ROIs because they change the object configuration. Of course they could be removed with function Screen ROIs as has been done in the Presence Verification chapter. But there is a simpler way here: open the parameter dialog of Create ROIs by Thresholding, choose **Options** and set the **Minimum perimeter** to 100. Function Check allowances will now report an error due to the reduced size of the visible portion of the drive plate and the reduced distance to the slider.



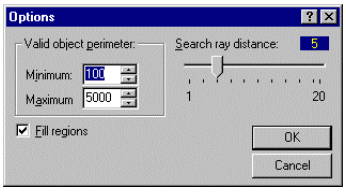
❼ NeuroCheck displays a graphical representation and a list of the measurements..

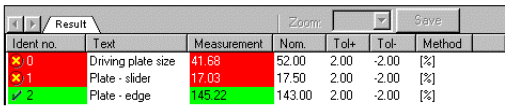


❽ The current measurements are supplied as defaults for the nominal value. Note the negative values entered for the lower allowance.



❾ All measurements are within the allowances.





❿ Results for image `disk-drive-dislocated.bmp`.

5.2 Change of Target Values in Automatic Mode

Problem

Changes in quality standards on production process could have the consequences that nominal values and allowances need to be adapted.
These changes often have to be made during the production process without switching into the manual mode of NeuroCheck.

Result

You know how to activate the permission to change parameters and target values in the automatic mode of NeuroCheck.

Solution

❶ Use the check routine of the previous section. We want to change the quality requirements for the floppy disk production. The allowances for the measurement **plate-slider** should be made editable in the automatic mode.

❷ Select **Data Input ▸ Parameters/Target Values Input Configuration** from the **Check Routine** menu. In the next dialog, select the check function Check allowances in the single check Driving plate concentric. Select **Publish**.

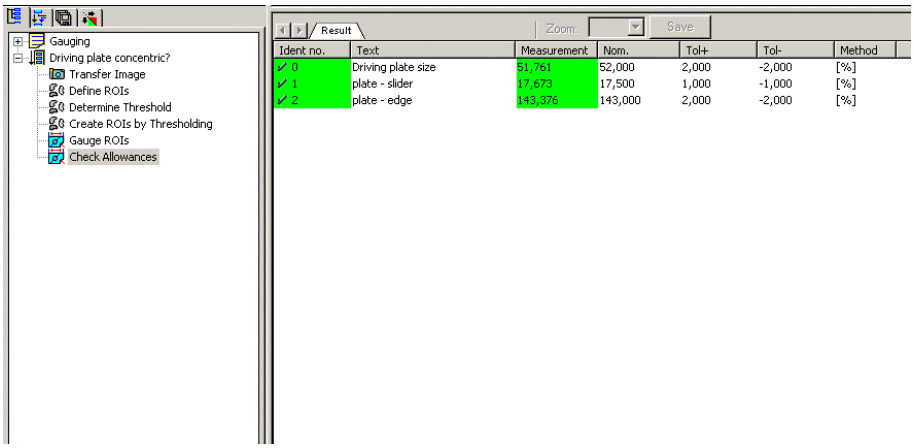
❸ Activate the check box **Upper Allowance** of the entry **plate-slider** and enter:

- in the field **user-defined text**: "% upper allowance slider-plate distance".
- in the field **Instructions / comment**: "For a high quality requirement choose a value close to 0."

Activate also the check box **Lower Allowance** of the same entry and enter:

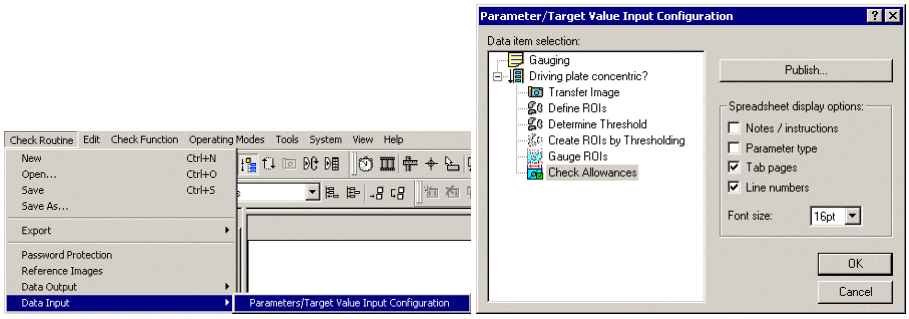
- in the field **user-defined text**: "% lower allowance slider-plate distance".
- in the field **Instructions / comment**: "For a high quality requirement choose a value close to 0. (Use negative values!)"

Confirm with **OK**.



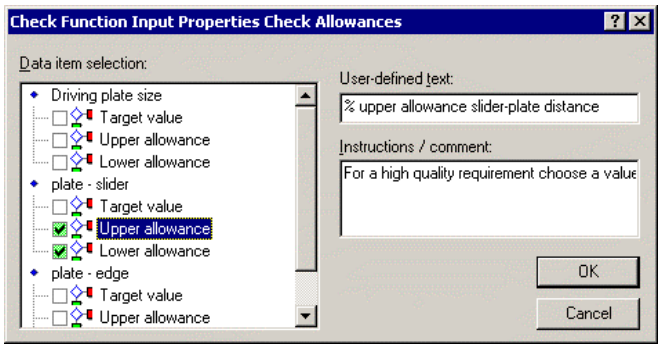
❶

Use the check routine from the previous section.



❷

Open the configuration dialog for the data input and choose Check Allowances.



❸

Activate the check box for the **Upper/Lower Allowance** of the measurement **plate-slider** and enter the text.

❹ You can see that the icon of the check function Check Allowances is marked with a small, green symbol. This shows that one or more values of this check function are editable in the automatic mode.

Activate additionally the check boxes **Notes / instructions** and **Parameter type** to get more information in the automatic mode when editing these target values.

❺ Choose **Remote Control** in menu **System** and select **Start check** on page **Input Signals**. Press **Edit** and select **Timer** as the input source, so that NeuroCheck restarts the check automatically. **Options** opens a dialog, where you can enter a time interval for restarting the check in seconds (e.g. 1 sec).

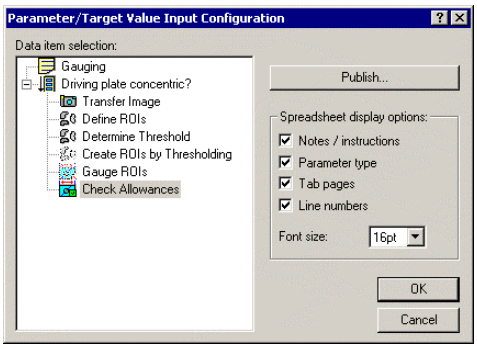
After switching to automatic mode, NeuroCheck will repeatedly perform the inspection run until you switch back to manual mode.

❻ Select **Parameter/Target Value Dialog** in the menu **Tools** or press the symbol in the toolbar. This opens the dialog **Parameter/Target Value Dialog,** Please note that the inspection continues in the background.

❼ Select the entry ”% upper allowance slider-plate distance“. Change the value `+2.00` to `+0.90`. Press **Apply.** NeuroCheck will use the new value for the allowance check in the next inspection cycle. Now the measured distance does not meet the specified allowances. The floppy disk doesn’t fulfil the new quality requirements and all inspections will return ‘not O.K.’.

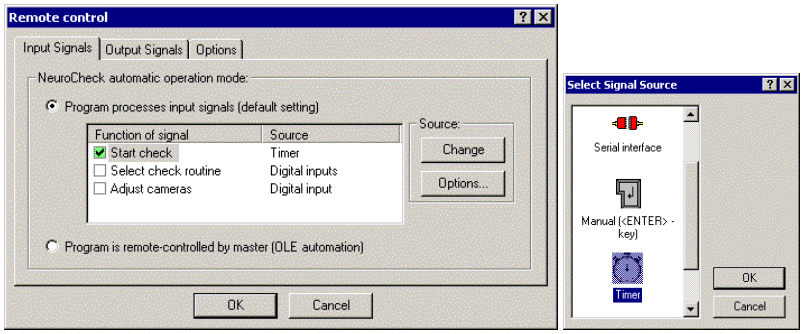
Remark

Not all parameters or target values can be made editable for the automatic mode. The criteria was that the basic logic of the check routine must not be changed. For instance, it is possible to configure nominal values and allowances, but it is not possible to determine if a measurement is checked at all. This decision is left to the application engineer who configures the check routine in manual mode.



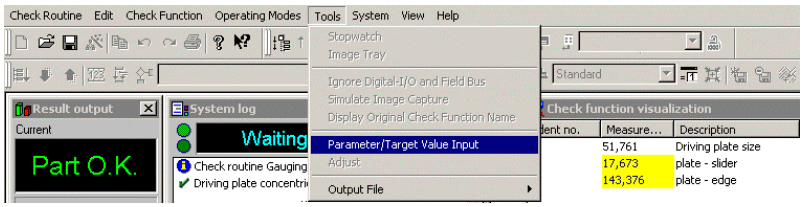
❹

Activate the additional display options.



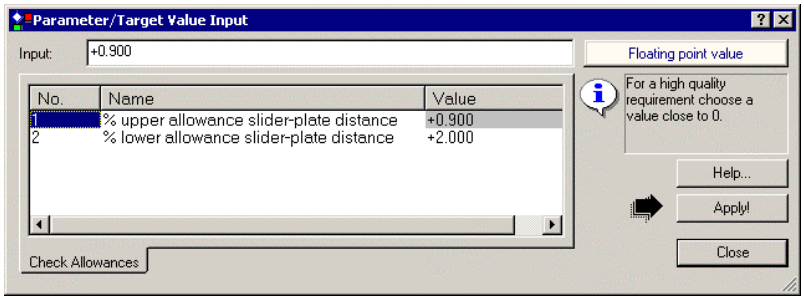
❺

Open the dialog from the menu **System ▸ Remote Control** and change the source of the start check signal to **timer**.



❻ 

Open the **Parameter/Target Value Dialog** by pressing the symbol during the automatic mode of NeuroCheck.



❼

After the input of the new value, you have to select **Apply** to activate the new upper allowance.

Remark: This dialog window is resizable.

5.3 Gauging Parallel Edges

Problem

You want to ascertain that the sliders on your disks are correctly mounted and closed. A correctly mounted slider is oriented parallel to the edges of the disk. The distance between a closed slider and the edges of the disk will be within a certain range.

Result

You know how to create object edges as self-contained objects, approximate them by straight lines and gauge their orientation relative to each other.

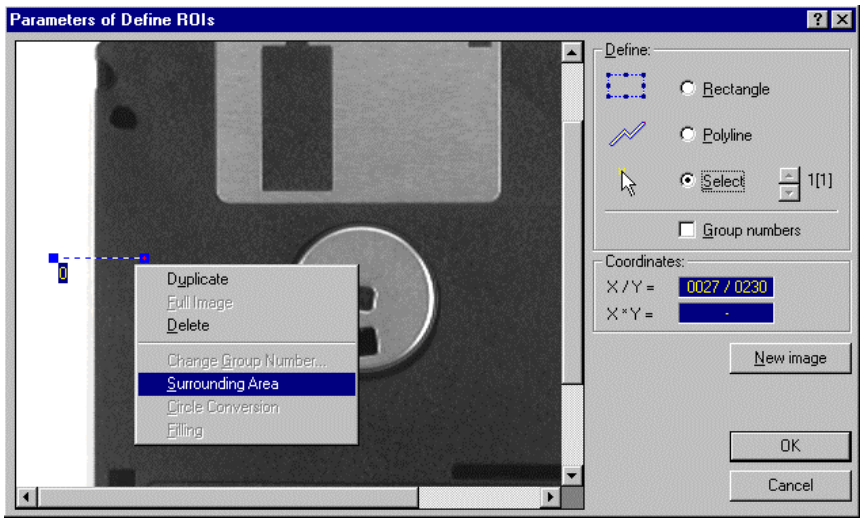
Solution

❶ Copy the check configured in the previous section. Remove the two last check functions, Gauge ROIs and Check allowances. First open the parameter dialog of function Transfer Image and select again the image `disk-protected.bmp`. Then open the parameter dialog of function Define ROIs.

This check has to establish parallelism and distance of edges. For this the respective edges have to be created as individual objects. The edges are searched for along linear regions of interest.. Open the context menu in the graphics panel and choose **Delete All**. Select **Polyline** drawing mode and draw a line from the background across the left edge of the disk holding the left mouse button down.

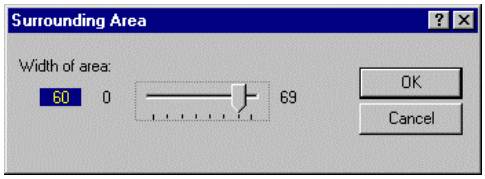
❷ Switch to **Select** mode and click the line with the right mouse button. From the context menu choose **Surrounding Area**. Use the slider in the dialog box to set as wide a surrounding area as possible (the possible width is computed dynamically from the distance of the polyline to the edges of the image).

❸ If the surrounding area extends over the limits of the disk, you have to reduce the size of the area or move the line further to the middle of the disk edge in order for the edge to be created correctly. To reduce the size, open the context menu again, choose **Surrounding Area** and reduce the area width using the slider. To move the line, click one of its end points with the left mouse button, keep the mouse button down and drag the line across the image.



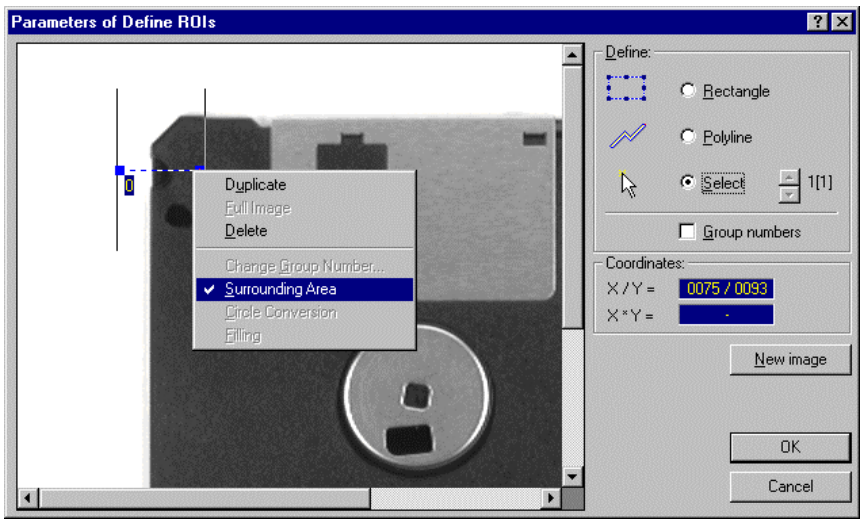
❶ 

Draw a line in **Polyline** mode. Switch to **Select** mode and choose **Surrounding Area** from the context menu.



❷

Set a surrounding area as wide as possible.

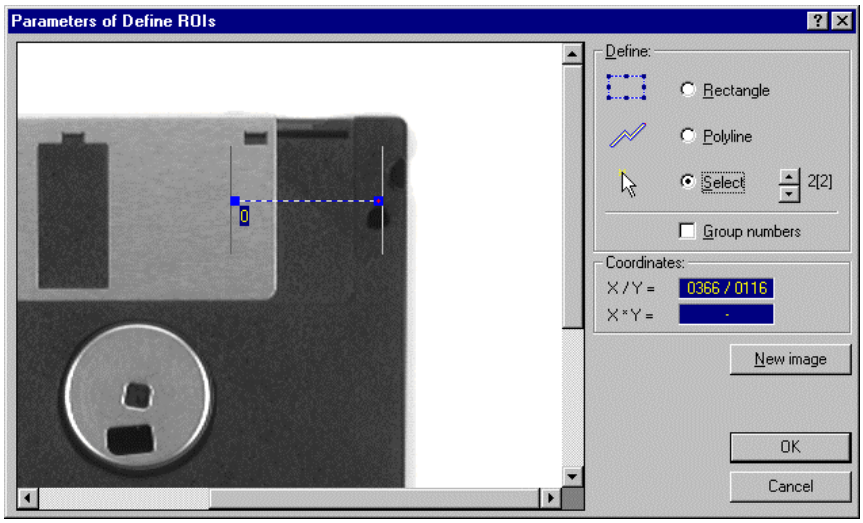


❸

Make sure that the surrounding area does not extend over the limits of the disk border, either by moving the line or by readjusting the width of the area.

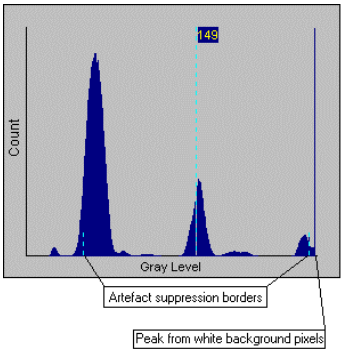
❹ Proceed in the same manner on the right edge of the slider: draw a line from inside the slider across its edge onto the disk casing. Make sure that the line ends outside the depression in which the slider moves in order to find the edge even if the slider should be opened completely. Set as wide a surrounding area as possible, but do not extend it over the straight portion of the slider's edge..

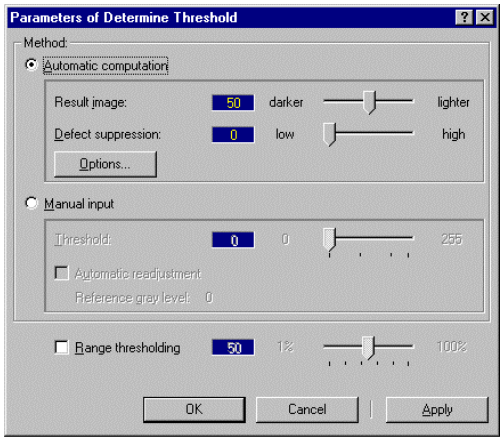
❺ When executing function Determine Threshold you will notice that the slider appears heavily structured in the thresholded image. A look at the histogram reveals the source of the problem: the threshold is computed from the distribution of gray levels inside all defined regions of interest including the surrounding areas. The pixels fall roughly into three groups: a lot of dark pixels from the disk corpus, a cluster of medium bright pixels from the slider and a smaller cluster of very bright pixels from the background inside the surrounding area of the polyline crossing the left edge of the disk. The automatic suppression of reflexes removes extreme gray levels from the computation. This is indicated by the small dashed lines near the borders of the histogram. In this case the peak at pure white is not an artifact but meaningful information that should not be excluded. To remedy this problem, open the parameter dialog of the function and move the **Defect suppression** slider to 0. Now the segmentation will work much better (of course it would be possible simply to move the threshold, but the automatic computation can compensate for a considerable variance in illumination).



❹

Make sure that the surrounding area does not extend over the straight portion of the slider's edge.



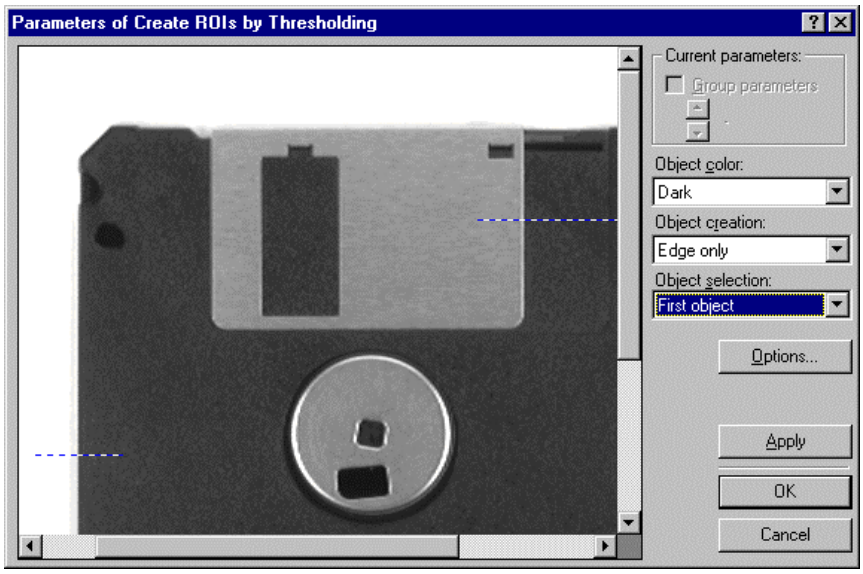


❺ 

Set the **Defect suppression** to 0 to avoid excluding the background pixels from the threshold computation.

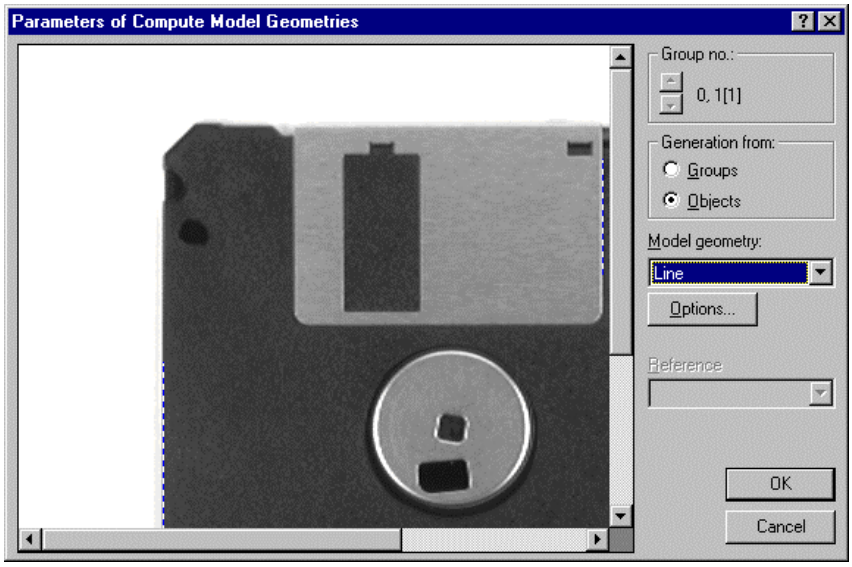
❻ In the parameter dialog of function Create ROIs by Thresholding choose object color **Dark**, object creation **Edge only** and object selection **First object**. This setting causes NeuroCheck to search for the first edge crossed by the linear region of interest. The portion of the edge lying within the surrounding area is created as an object. After executing the function NeuroCheck displays the two created edge sections as objects, but due to their small width they will not always be easy to see.

❼ The edges are no ideal straight lines in a mathematical sense as is required for evaluating parallelism. Therefore, the next step is to approximate the edges by ideal straight lines. Append function Compute Model Geometries from the **Gauging** page. In the parameter dialog select **Generation: from objects** and **Model Geometry: Line**. The effect is that NeuroCheck computes a straight line for each of the edges using a least squares fit algorithm.



❻ 

Set the search parameters to **Dark**, **Edge only** and **First object**.



❼ 

Choose generation of a line as geometrical model for individual objects.

❽  

In principle you proceed further as in the previous section. Append function Gauge ROIs, open its parameter dialog, choose **New** and select both objects. Choose **Line** as the geometrical description for the first object, **Point** for the second. On the second page choose **Distance, absolute** as the gauging rule. This rule computes what is defined as the distance between a point and a line in mathematics: the length of a line from the **Point** to an extension of the **Line**, perpendicular to the direction of the computed line.

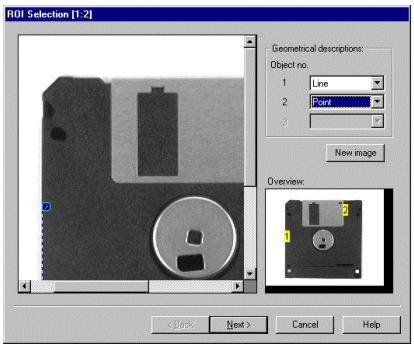
❾ 

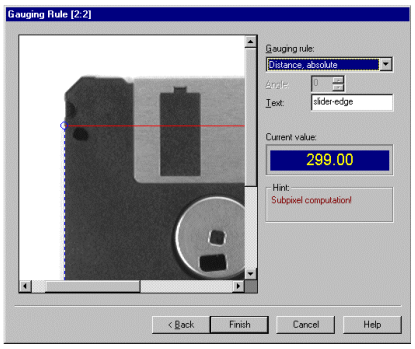
Again choose **New** and select both objects, but this time use **Line** as the description for both objects. On the second page select **Intersection, angle** as the gauging rule and choose **Finish**. After execution NeuroCheck displays the distance as a solid line, together with a dashed auxiliary line indicating the extension of the edge. The angle measurement is indicated by a circle at the disk edge.

❿  

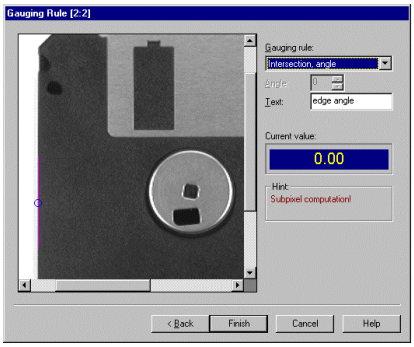
As in the previous section append function Check allowances and set nominal values and allowances. In this case we can give a correct setting for the intersection angle without further experiments: the nominal value has to be 0° for exact parallelism and a deviation of ±1° can be tolerated. Of course it would not be meaningful to use a percentage for the allowances, when the nominal value is 0, so this check has to be done absolute.

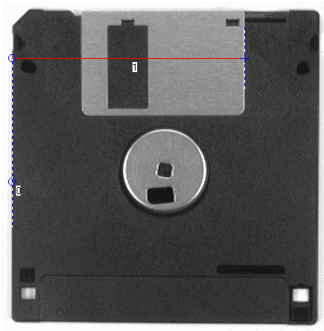
With these two measurements, you can now check, whether sliders are open or damaged, as you can easily verify using images `disk-slider-open.bmp` and `disk-slider-dislocated.bmp`.



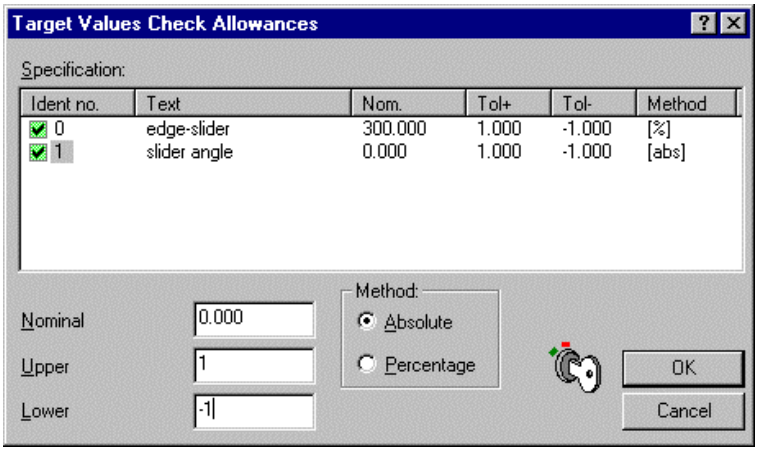


❽ Using the **Line** as geometrical description for one of the objects, the **Point** for the other one, the perpendicular distance between the two straight lines can be measured.

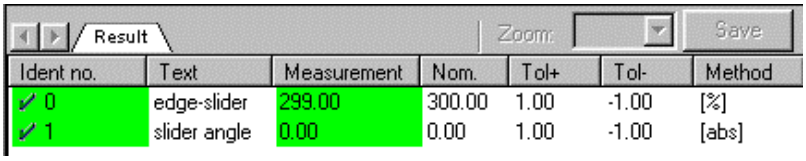




❾ NeuroCheck indicates the angle measurement by a small circle on the disk edge, the distance by a solid line and a dashed auxiliary in the direction of the edge.



❿ Enter nominal values for the measurements.



Both measurements comply with the allowances.

5.4 Output of Measurements to a File

Problem

You want to carry out statistical calculations with measurements, e.g. to evaluate the average precision of the mounting of the sliders.

Result

You know how to write measurement values from NeuroCheck into a file that can be read by other programs.

Solution

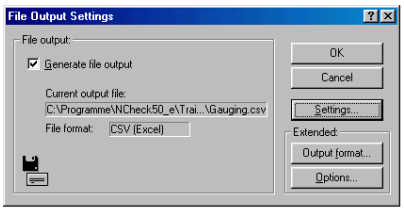
❶ Use the check routine created in the previous sections of this chapter. From the **Check Routine** menu choose **Data Output ▸ File...**. As soon as you have activated the **Generate file output** check box in the **File Output Settings** dialog various buttons become available. First choose **Settings...** to enter the **File Output Wizard**.

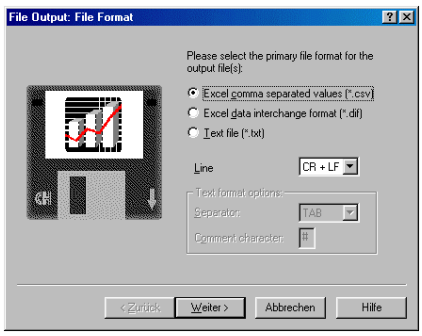
On the first page you select the basic file format. The preselected CSV (comma separated value) format is a standard data exchange format understood by most spread-sheet programs, e.g. Microsoft Excel. Simply click **Next** here.

❷ On the **Output Mode** page you can select between writing to a single file or to a file series. Again click **Next** here. Click **Browse** on the next page.

❸ In the file select dialog, enter a file name, e.g. “Gauging”, and confirm with **Open**. The click **Next** to step on to page **Comment**. Activate the **Write additional information ...** check box. Now you can enter a comment in the edit box below which will be written at the beginning of the output file to help with using and evaluating the output file. After leaving the wizard with **Finish**, the **File Output Settings** dialog will display the select file name – possibly abbreviated – and the file format.

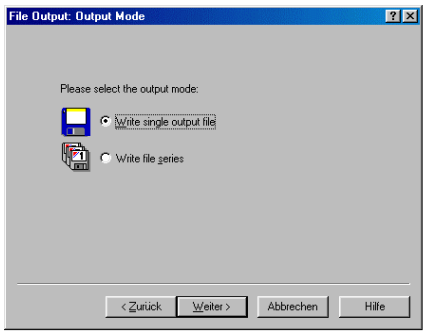
Upon leaving the **File Output Settings** dialog box the output file is created using several default settings for generating meaningful output without further configuration. All settings can be changed using the **Output format** button in the **File Output Settings** dialog, except for the regional settings for numerical and date formats which are always taken from the current Windows configuration.

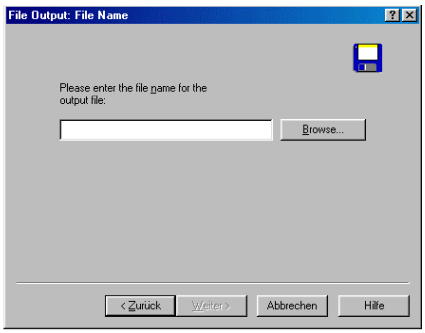




❶

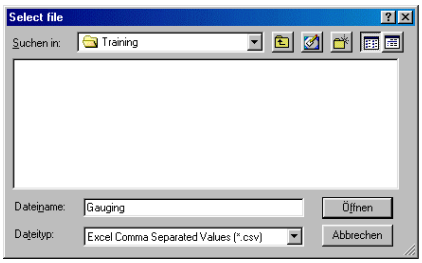
Activate output of data to file and select the default CSV format.

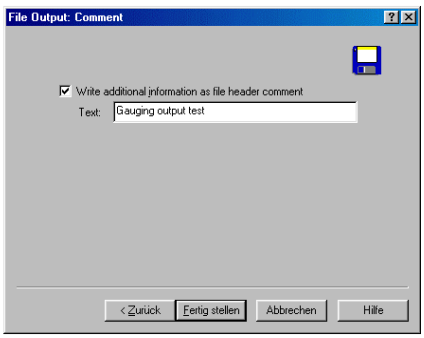




❷

Configure output to a single file.





❸

Enter a file name and activate output of additional information into the file header.

❹ 

Switch to the tabbed edit page **Output**. You will see that the check box **Destination: result file (global)** directly below the check routine name is now activated. It corresponds directly to the **Generate file output** box in the **File Output Settings** dialog. Deactivating this check box cancels all file output without requiring any changes to the configuration of individual check functions.

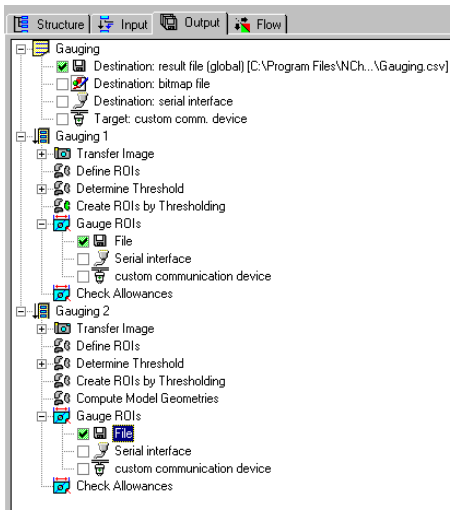
Now activate the **File** check boxes below function Gauge ROIs in both checks to tell NeuroCheck that you want the result values of these two functions to be written to the file.

❺ 


Switch to automatic configuration mode. Click the existing check function visualization window with the right mouse button and choose **Parameters**. In the **Parameters** dialog activate the visualization of function Gauge ROIs from the first check and choose **Measurement values** from the **Display** list. Then add another check function visualization window by clicking the corresponding button in the screen configuration toolbar and drawing the window frame with the mouse. Proceed as before, but activate the checkbox of function Gauge ROIs from the second check. Now order the windows approximately as displayed on the opposite page.

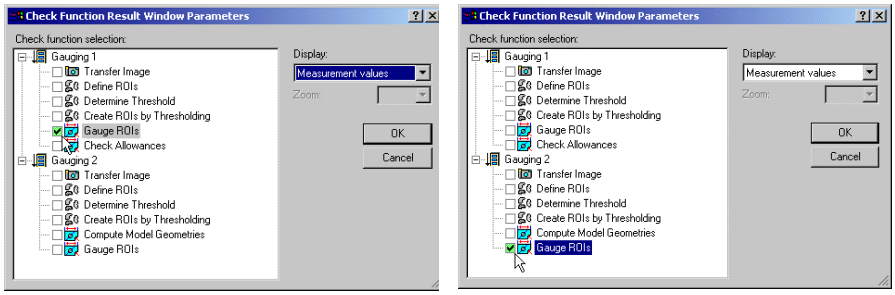
❻ 

Switch to automatic mode and press **enter** a few times to execute the check routine (if NeuroCheck does not react, you may have to choose **Remote Control** from the **System** menu, activate the **Start Check** signal on the **Input Signals** page of the **Remote Control** dialog and choose **Change** to select **Manual** as the signal source). If you are working directly from a camera image, you can switch the disks in between to observe variations in the values, but you have to take care to position the disks precisely. If you are working from bitmap files, you can use **Browse** in the parameter dialog of function Transfer Image and select several bitmap files using the standard multiple selection methods of Windows (dragging the mouse or pressing **Ctrl** or **Shift** together with the left mouse button). NeuroCheck continues to count the inspection runs and to append new values to the existing output file.



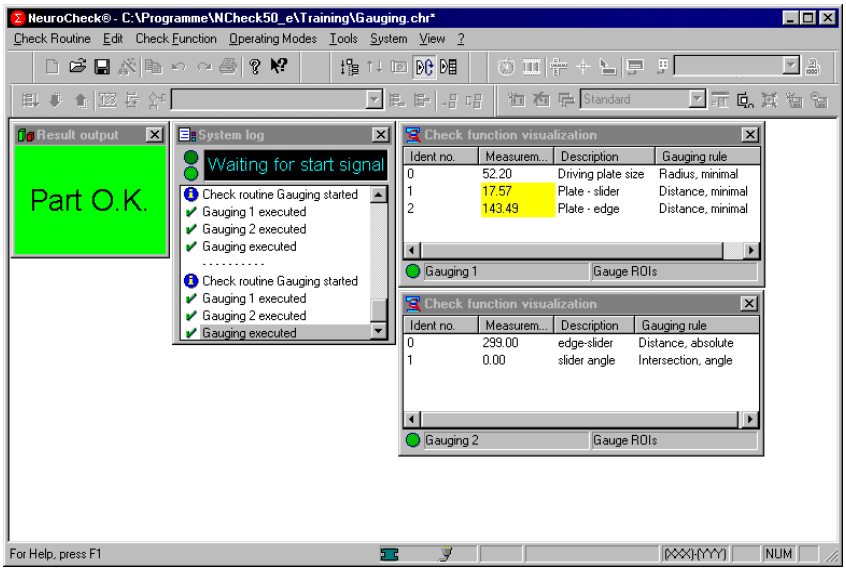
❹

Activate data output to file for function Gauge ROIs in both checks.



❺

Activate the check function visualization windows of the two gauging functions.



❻

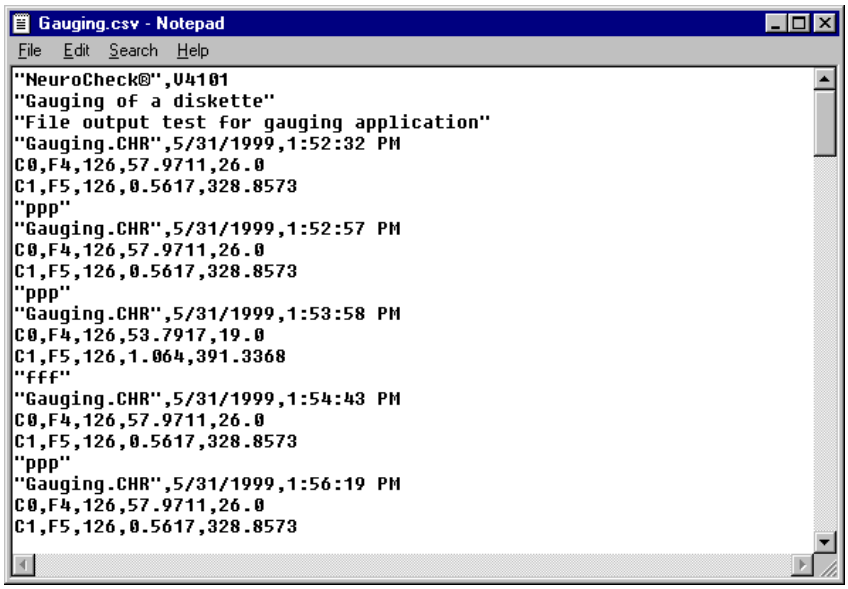
Automatic mode screen layout.

❼ Open a text editor, e.g. “Notepad” (you find this under **Accessories** in the Windows™ 95/NT start menu). Load the CSV file just created (`Gauging.csv` in this example). The file starts with a three line header:

1. The first line identifies the program, which created the file: “NeuroCheck”, followed by the version number of the file format. This version number can be used, for example, by an evaluation macro in a spreadsheet program to determine whether it can process this file correctly.
2. The second line contains the description text of the check routine entered in the **Check Routine Properties** dialog box.
3. The third line consists of the File info you entered in the file selection dialog.

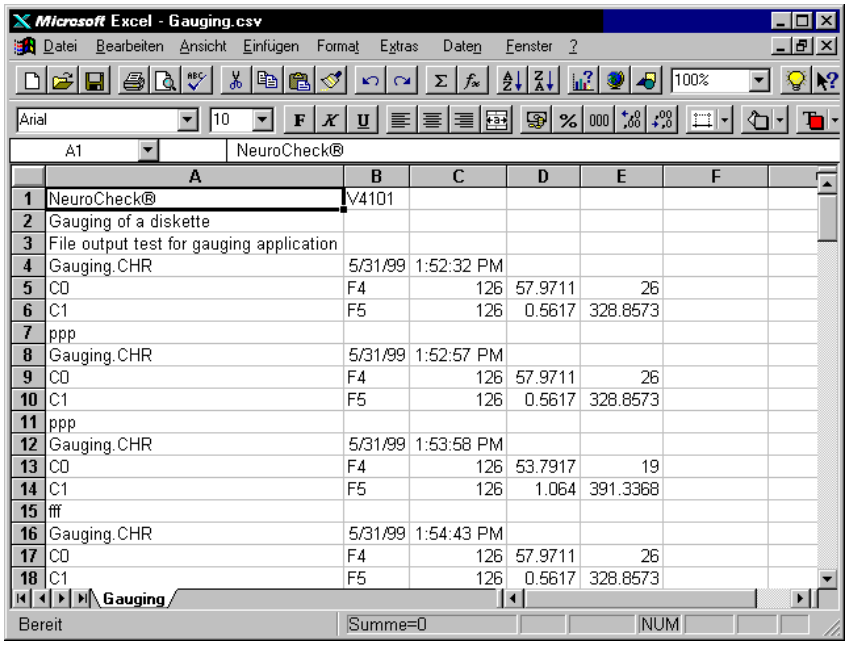
NeuroCheck writes one block of data per inspection run. Such a data block consists of the results of the check functions, the check routine info, and the check result. The results of each check function are written in a single line starting with the check index (indicated by a C, and counted from 0), followed by the check function index (indicated by an F, counted from 0), the check function identification number (526 for check function Gauge ROIs) and the measurement values. The **Options** button in the **File Output Format** dialog allows for detailed configuration of all these individual information items. The check routine info contains the file name of the check routine and date and time of the inspection run. The check result is represented by “ppp” (passed) or “fff” (failed).

❽ If you have a spread sheet program available, you can use it to view the file `Gauging.csv`. Please note that Microsoft Excel may behave differently, depending on its configuration and the method used for opening the file (double-click, drag & drop or menu command). Sometimes regionally specific settings for number or date format may not be recognized correctly. If the file is opened using the appropriate menu command, Microsoft Excel will use the values from the Windows registry which are also used by NeuroCheck.



❼

The CSV file viewed in a text editor.



	A	B	C	D	E	F
1	NeuroCheck®	V4101				
2	Gauging of a diskette					
3	File output test for gauging application					
4	Gauging.CHR	5/31/99	1:52:32 PM			
5	C0	F4	126	57.9711	26	
6	C1	F5	126	0.5617	328.8573	
7	ppp					
8	Gauging.CHR	5/31/99	1:52:57 PM			
9	C0	F4	126	57.9711	26	
10	C1	F5	126	0.5617	328.8573	
11	ppp					
12	Gauging.CHR	5/31/99	1:53:58 PM			
13	C0	F4	126	53.7917	19	
14	C1	F5	126	1.064	391.3368	
15	fff					
16	Gauging.CHR	5/31/99	1:54:43 PM			
17	C0	F4	126	57.9711	26	
18	C1	F5	126	0.5617	328.8573	

❽

The CSV file viewed in Microsoft Excel.

5.5 Using Metrical Units

Problem

In a real-world gauging application you will of course determine metrical measurement values instead of the pixel measurements used throughout this chapter.

Result

You can determine the conversion factor between image pixels and real-world dimensions and know how to inform NeuroCheck of this conversion factor.

Solution

The basic solution is very simple. You need a measurement that can be determined reliably, have NeuroCheck compute this measurement in pixels, divide the metrical measurement by the pixel count and use function Calibrate Pixel Size to set this conversion factor. A typical method is to perform this calibration in a separate check and switch this check off during automatic execution.

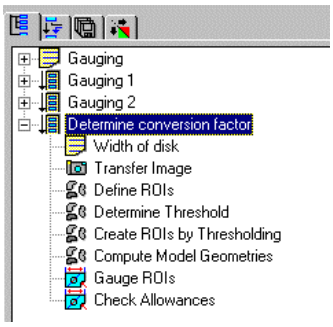
❶ 
Use the check routine from the previous section and copy the second check in which the distance from the edge of the disk to the edge of the slider has been measured. Rename the check, for example to “Determine conversion factor”.

❷ 
In the copied check open the parameter dialog of function Define ROIs and delete the line crossing the edge of the slider. Define a new line crossing the right edge of the diskette, starting on the background. This way NeuroCheck can search for a dark edge on both lines and you do not need to configure the check to search for dark and light objects simultaneously. Set a surrounding area for the line similar to the line across the other edge of the diskette. The following functions already do what is needed here: both edges are created as objects and for each of the edges a line is computed as the model geometry.

❸ 

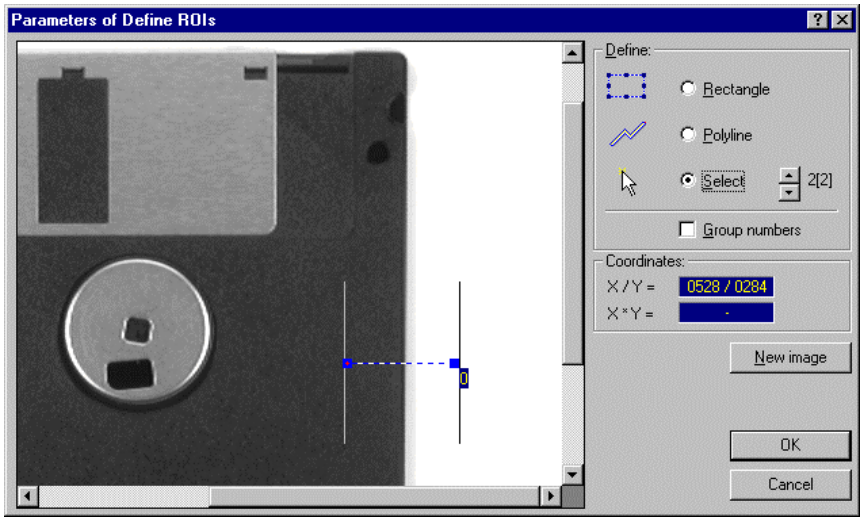
Now open the parameter dialog of function Gauge ROIs. In the original configuration of the previous check two measurements have been defined: the absolute perpendicular distance between the two edges and their intersection angle. Select the **Intersection, angle** measurement, then choose **Delete**. Now only the absolute distance of the two edges remains. Choose **Edit** switch to the second page of the **Gauging Wizard** and change the descriptive text, for example in “Width of disk”. Leave the wizard and the parameter dialog with **OK** and execute the function. In this example it yields a value of 393.43 pixels. A 3.5" disk has a width of 88.9mm. This means that an image pixel has a dimension of 0.226 mm.

The last function, Check Allowances, can now be deleted.



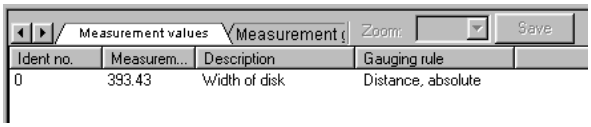
❶ 

Copy and rename the second check.



❷ 

Delete the line crossing the edge of the slider and draw a new line from the background across the right edge of the disk. Set a surrounding area for this line.



❸ 

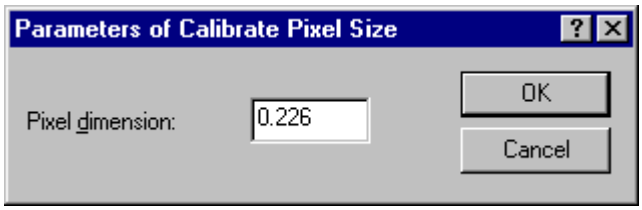
After deletion of the angle, Gauge ROIs yields the width of the disk in pixels.

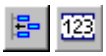
❹ Open the first check of the check routine. Insert function Calibrate Pixel Size from page **Tools** before the first function of the check. Open its parameter dialog and enter the pixel dimension of 0.226 mm.

❺ When you now execute the check up to function Gauge ROIs NeuroCheck will display the measurement values in millimeters.

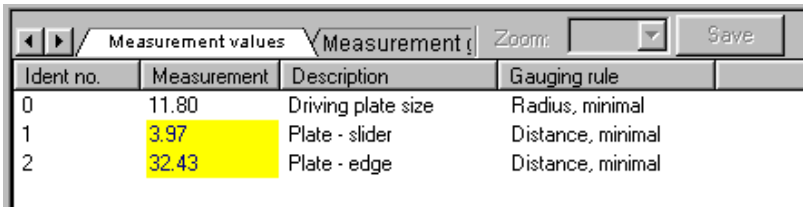
❻ Executing the check one step further, to function Check Allowances, yields 'not .O.K.' because this function still uses the manually entered nominal values from before. Open its target value dialog and convert the measurements using the pixel dimension. In a real-world application you usually would not need to do this because in that case the nominal values would be given in metrical units beforehand.

❼ You can now treat the second check in the same manner: insert Calibrate Pixel Size at the beginning of the check, execute it up to Gauge ROIs, and convert the nominal values of Check Allowances – with the exception of the angle, of course.



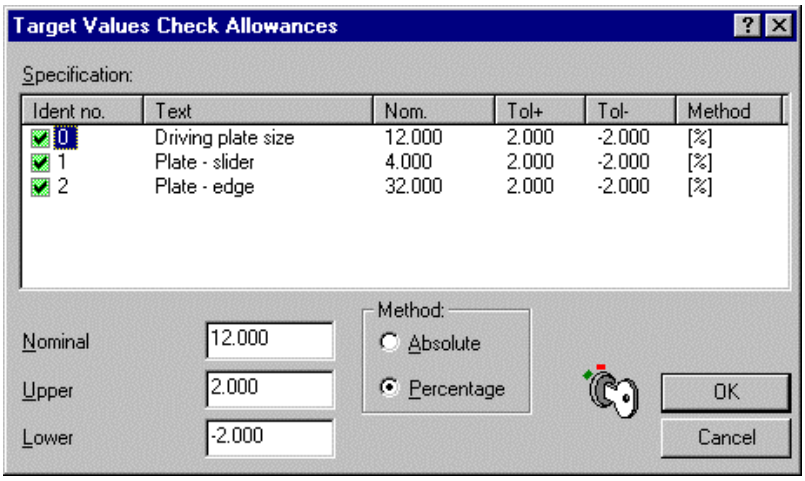
❹ 

Insert function Calibrate Pixel Size at the beginning of the check and enter the pixel dimension.



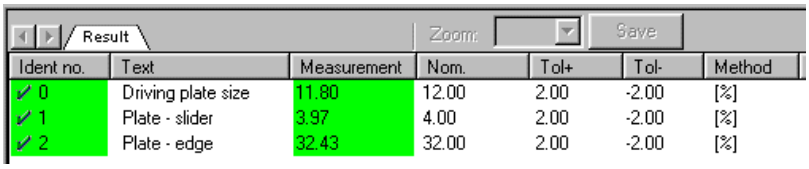
❺ 

Executing Gauge ROIs yields the measurements in millimeters.



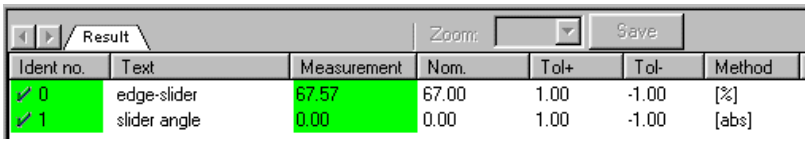
❻ 

The nominal values have to be changed manually. In a real-world application they would usually be given in metrical units beforehand.



❼

Results of Check Allowances in millimeters – except for the angle, of course.

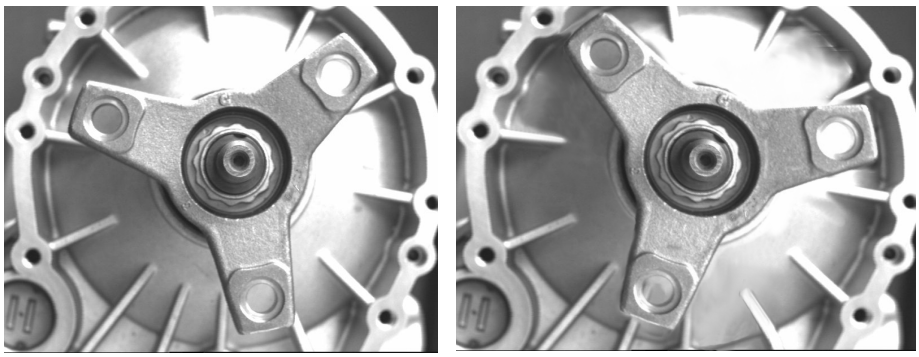


6 Position Adjustment

Usually it is not possible to achieve absolutely precise positioning of objects in front of the camera. Sometimes the hierarchical object search cannot be applied to compensate for position variations. Especially gauging applications usually require more control about the positioning of objects like search lines. For such cases NeuroCheck offers an automatic position adjustment.

In this chapter you will learn

- how to compensate for translation of objects,
- how to compensate for rotation of objects,
- how to transmit positions and orientations via serial interface, e.g. to control robots used for handling the objects.



Flange in two different positions

6.1 Compensating for Translations

Problem

In the gauging application from section 5.1, **Measuring Object Distances**, the positioning of the disks below the camera cannot be guaranteed. The hierarchical object search of section 4.2, **Presence Verification Using Object Properties,** cannot be used here because the slider does not lie within the limits of the disk casing object and merges with the background. Furthermore function Gauge ROIs requires a defined arrangement of objects.

Result

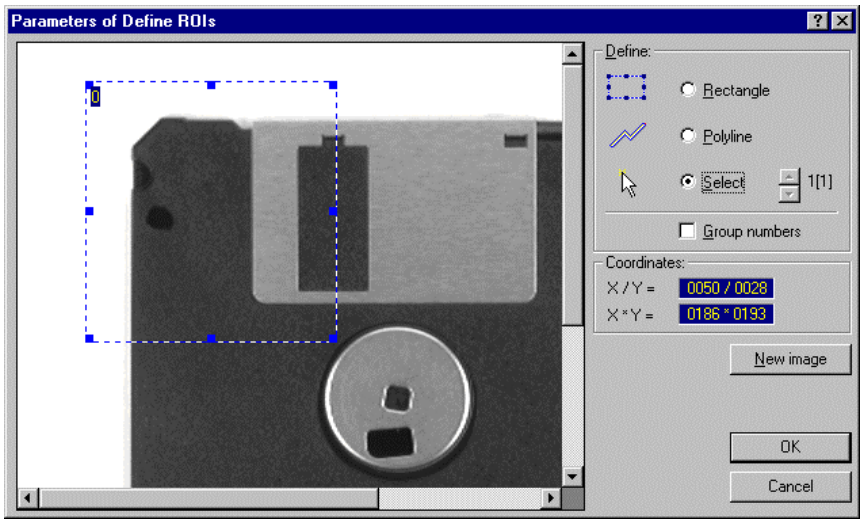
You are able to configure a check routine with automatic position adjustment, compensating object translations in horizontal and vertical directions.

Solution

❶ Create a new check routine. Append function Capture image (only if you are working from camera) and Transfer image. Load a complete image of a disk like in the preceding sections. Append function Define ROIs, open its parameter dialog and define a rectangle partly covering the disk. Because NeuroCheck can always create the complete object even if only part of it lies within the search region, the rectangle does not have to enclose the disk completely. Size and position of the rectangle depend on the position variations to be expected, i.e. you have to ascertain that the disk will be found in every position it can possibly occupy in front of the camera. The disk being a fairly large object, this is not a severe restriction.

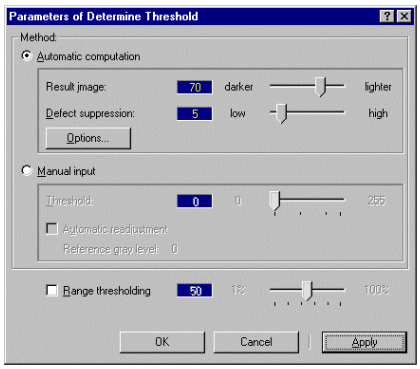


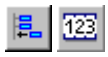
❷ Append function Determine threshold and adjust the brightness of the result image so that the slider and the casing of the disk exhibit clear contours and no structuring.

❶ 

Define a region of interest covering part of the disk. Size and position of the region have to be adequate to find the disk in every position it can possibly occupy in front of the camera.



❷ 

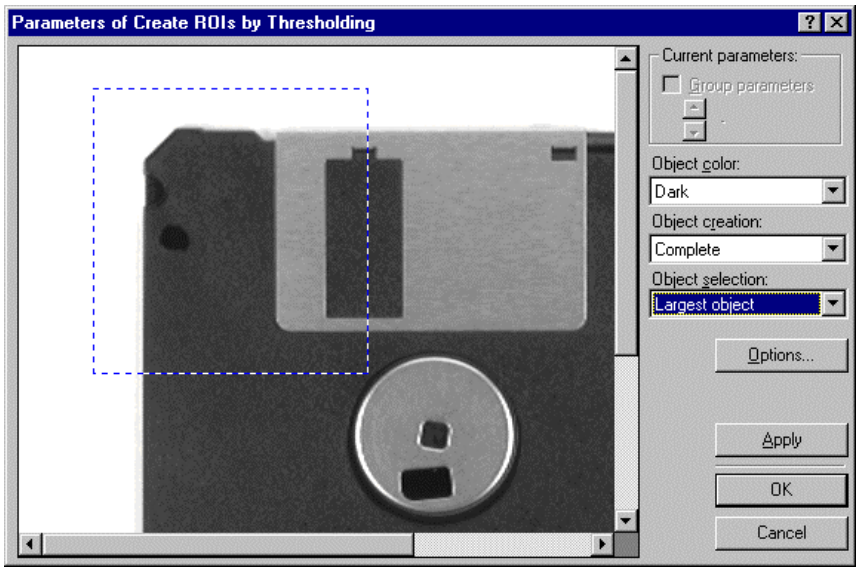
Adjust the brightness of the result image in function Determine threshold so that the diskette has clear contours and appears untextured in the binary image.

❸ 

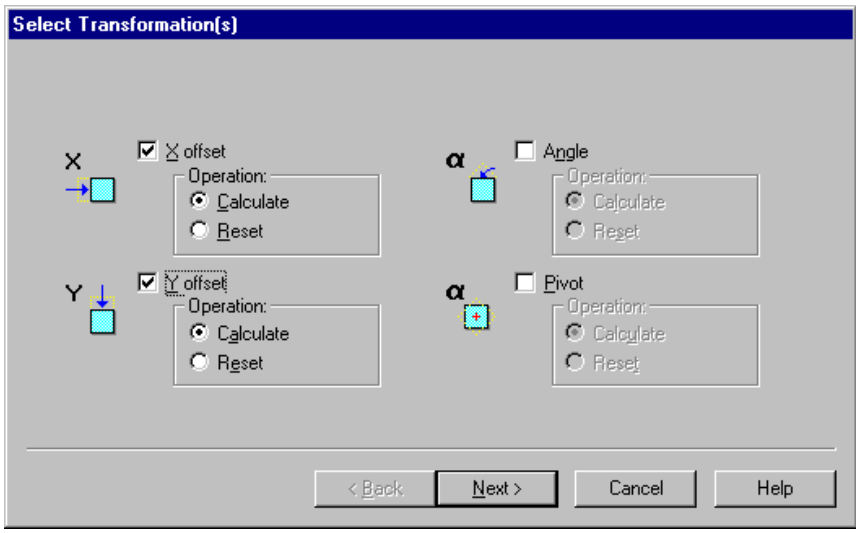

Now append function Create ROIs by thresholding. Take care that **Complete** is selected in the **Object creation** box and **Largest** in the **Object selection** box. This will create the complete disk, but only the disk and no other dark objects that happen to be inside the region of interest.

❹ 


Up to here it has been fairly standard procedure. Now comes the important part. The next function is Determine position of ROIs from page **Position**. Open the parameter dialog. It shows the disk object indicated by a dotted line but not much more, because no reference position has yet been defined. Choose **Teach...**. The first page of the **Positioning Wizard** appears. Activate the check boxes **X offset** and **Y offset**. The wizard automatically activates the **Calculate** buttons below, indicating that from now on it will compute offsets in x and y direction from a reference object. Choose **Next**.



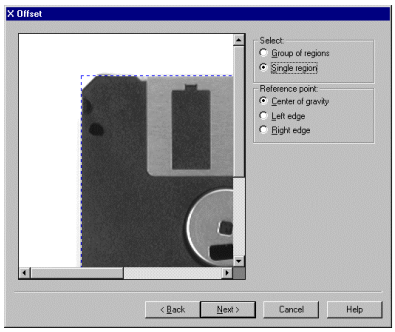
❸ Set object selection to **Largest** in function Create ROIs by thresholding.

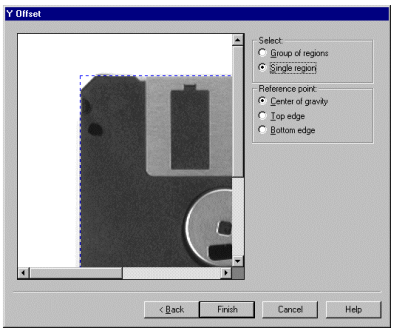


❹ Choose **Teach...** in function Determine Position to call the **Positioning Wizard**. On its first page, activate computation of x and y offsets.

❺ On the following pages you can simply choose **Next**, because there is only a single object, so you do not have to select one, and the default setting to compute the reference position from the center of gravity of a single object is correct for this application.

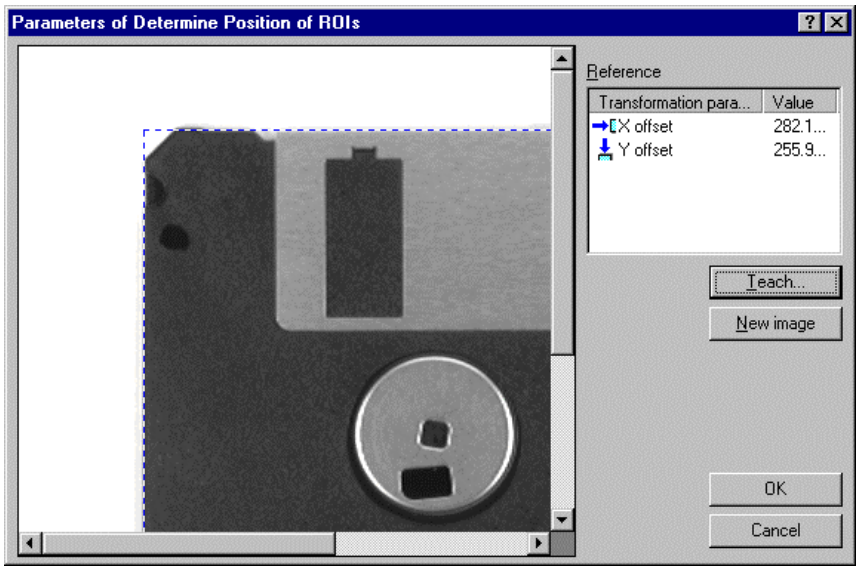
❻ After completing the wizard, you are back in the parameter dialog of function Determine Position of ROIs. Now it displays the reference coordinates for x and y.





❺

Choose **Next** on the following pages, because the default settings are already correct.

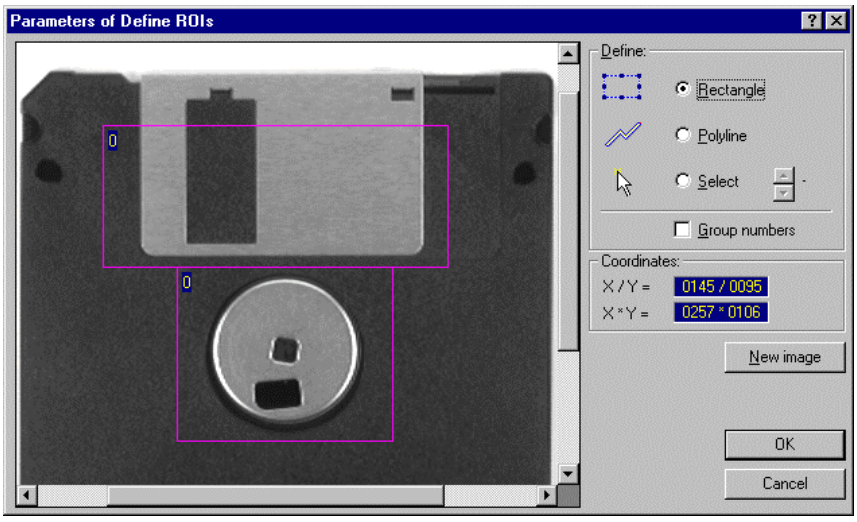


❻

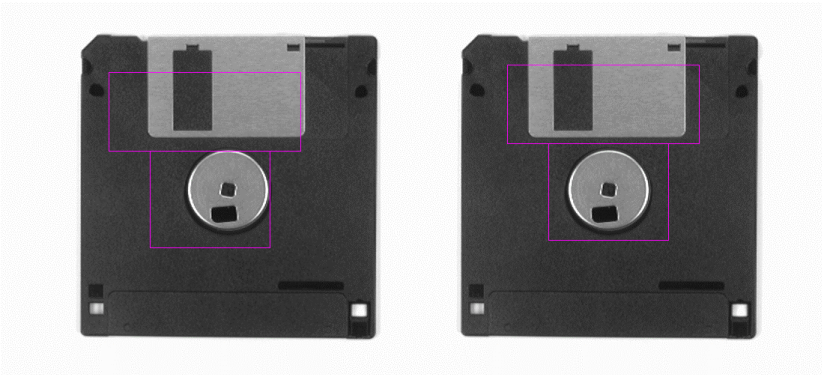
The parameter dialog of Determine Position with the reference coordinates computed by the wizard.

❼ Append function Define ROIs and define two rectangular regions, as in section 4.2, **Presence Verification Using Object Properties**, covering the complete drive plate and the lower portion of the slider.

❽ Next you need function Position ROIs, again found on page **Position**. You do not need to set any parameters for the time being. Executing the function results in – nothing, as was to be expected with a disk placed precisely in the reference position. Open the parameter dialog of function Transfer image and shift the image section in such a way that the disk still will be loaded completely but in a slightly different position within the image section (moving the disk in front of the camera is also possible, but it is quite difficult not to rotate it doing so, and the check does not yet include rotation compensation). Execute the check routine stepwise and you will see that the two rectangular regions of interest are incorrectly positioned after the second Define ROIs function and shifted to the correct position by function Position ROIs.



❼ Define two rectangular regions for the complete drive plate and the lower portion of the slider.



❽ On a shifted disk, the regions are at first positioned incorrectly then shifted to the correct positions.

6.2 Compensating for Rotations

Problem

In the previous section we assumed that the disk might be shifted horizontally or vertically, but not rotated. Now we also want to determine the rotation angle, although the disk as an almost quadratic object is ill suited for computing the orientation.

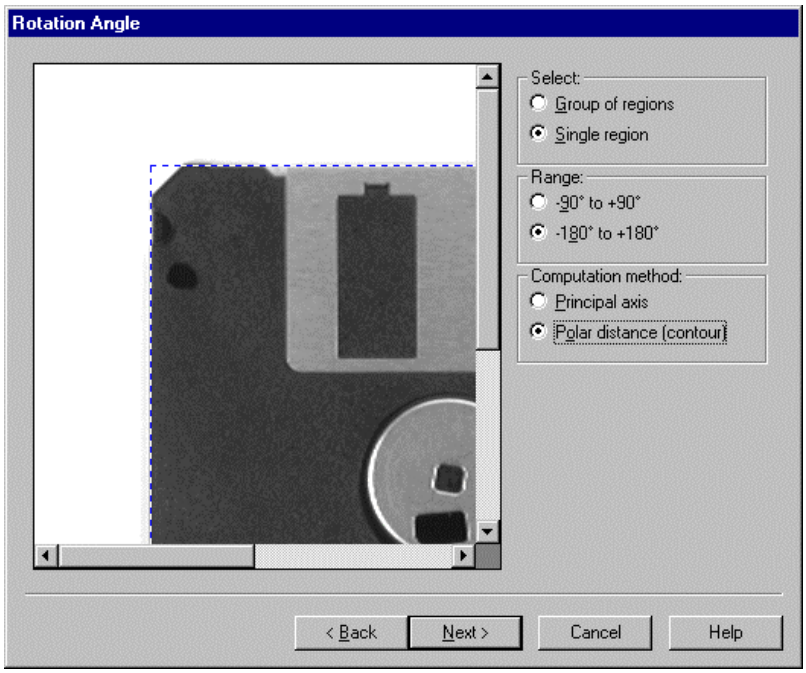
Result

You know how to configure the position adjustment functions to compensate for rotations and you know how to determine the orientation of difficult objects.

Solution

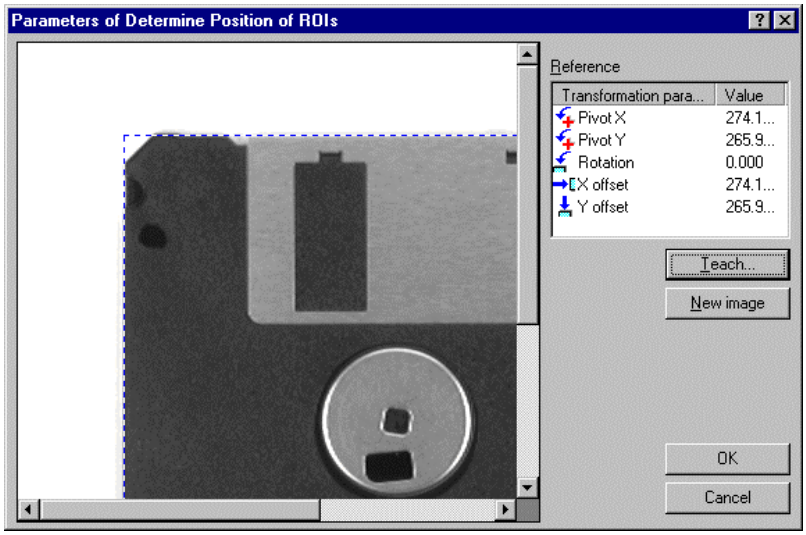
❶ Copy the check from the previous section and open the parameter dialog of function Determine position and choose **Teach...**. Activate all four check boxes on the first page of the **Positioning wizard**. On the first two pages you can simply choose **Next** as before without setting parameters. On the third page activate the **Polar distance** option. It is more computation intensive than using the principal axis, but for near circular or square objects like the diskette, the principal axis method does not give very stable results.

❷ Although you can simply choose **Next** on the **Pivot** page of the **Positioning Wizard**, it was nevertheless very important to activate **Pivot** on the first page, because otherwise the center of rotation would be the top left corner of the image, which would lead to surprising results in this case. The reason that NeuroCheck allows you to freely set the center of rotation for position adjustment is that there are applications, where an object outside the center of the scene is better suited for computing the rotation angle. For example, you might have a long asymmetrical object in the image, which allows using the fast **Principal axis** option for computing the angle, but have to rotate the regions around a different point. NeuroCheck gives you complete flexibility in this respect, but, on the other hand, this flexibility requires the pivot to be set explicitly.



❶ 

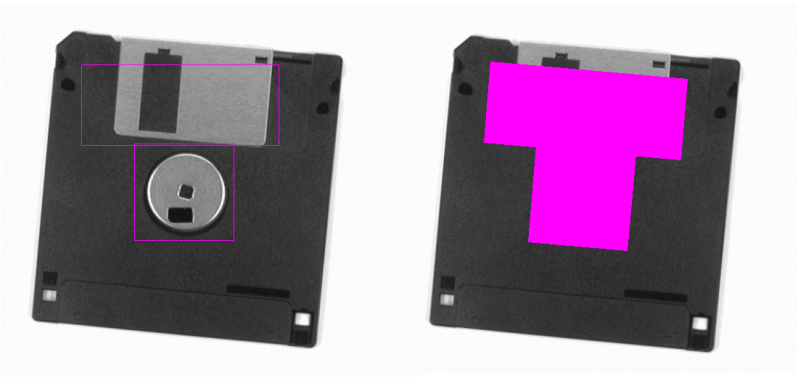
Set the computation method to **Polar distance** on the **Rotation** page.



❷

The parameter dialog now displays all five positioning parameters. Note the correspondence between the reference for the pivot and the offset in x and y: rotation takes place around the center of the diskette.

❸ If you now rotate the disk or select an image of a rotated disk and execute the check routine stepwise, you can observe the compensation of the rotation. The regions will then be drawn in solid color to indicated that they are no longer shifted rectangles but arbitrarily orientated regions.



❸ 

After rotating the regions appear in solid color to distinguish them from merely shifted rectangles.

6.3 Transmitting Position Parameters

Problem

The disk is to be handled by a robot. The robot needs the position parameters to find the disk and seize it.

Result

You know how to transmit position and orientation of a reference object to an external control device.

Solution

This section requires correct configuration of the serial interface, as explained in section 9.4 **Configuration of the Serial** Interface and a second computer (or a free communication port) as receiver.

❶ 

Switch to the **Output** tab page of the edit pane. Activate the check box **Destination: serial interface** in the global tree branch to instruct NeuroCheck to generate serial output at all. Then activate the **Serial interface** check box in the branch of function Determine Position in the second check.

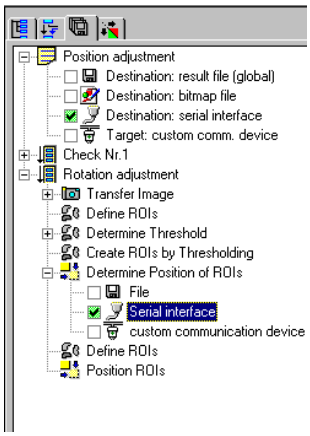
❷ 

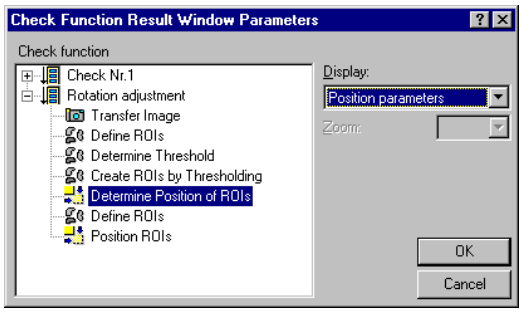
Switch to automatic screen configuration mode. From the context menu of the check function visualization window choose **Parameters**. Activate the output of function Determine Position in the second check and choose **Position parameters** from the **Display** list.

❸ 

Switch to automatic mode and run the check routine a few times. The visualization window and the screen of your terminal program display identical position parameters (apart from higher precision used for the serial communication).

The simple ASCII format of this output can easily be interpreted by any PLC, but to communicate with a PLC one would leave out the line feed characters, which improve the readability in the terminal program.



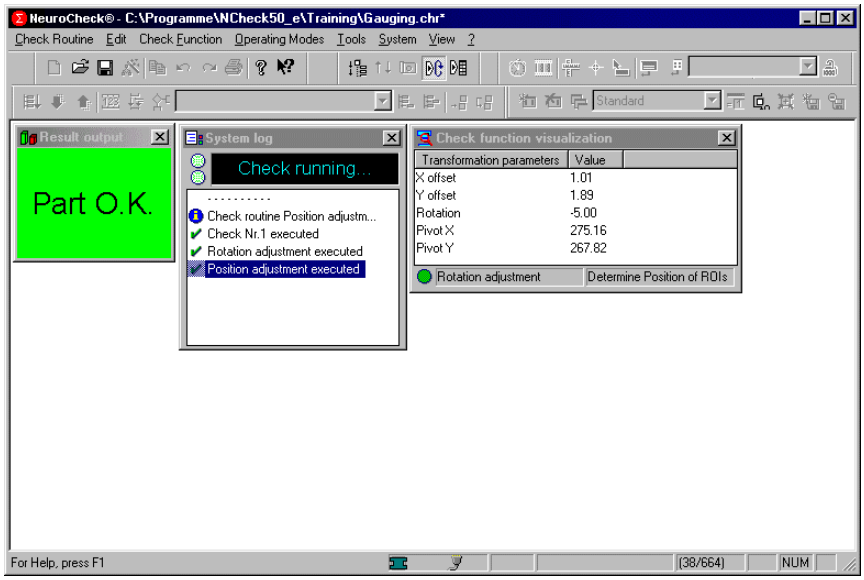


❶ 

Activate the **Serial interface** check box globally and for check function Determine position in the second check.

❷ 

Activate the output in the check function visualization window of function Determine Position from the second single check.



❸ 

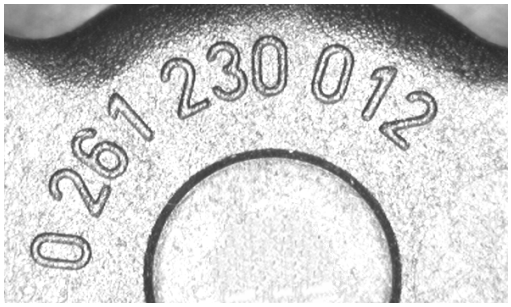
NeuroCheck writes the position and orientation to the measurement log and send them (with higher precision) via serial interface.

7 Optical Character Recognition

One of the most demanding applications in image processing is optical character recognition, OCR. NeuroCheck uses artificial neural networks for the identification of characters (and arbitrary symbols), because of their adaptability and their tolerance against noise and image degradations.

In this chapter you will learn

- how to segment the characters from the background,
- how to create a representation of the characters, that can be used as input to a classifier,
- how to generate training data for a classifier,
- how to train a classifier,
- how to set target classes for the character recognition,
- how to transmit read characters via serial interface.



Engraved numbers on a cast part

7.1 Creating Training Data

Problem

You want to compare a bar code with the plain text belonging to it.

Result

You know how to reuse a finished solution for a specific image processing problem and can configure a check for character recognition.

Solution

❶ 

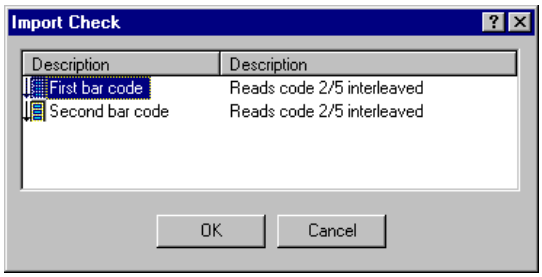
Create a new check routine. From the **Edit** menu choose **Import Check...**. In the file selection dialog open the check routine for bar code identification created using the check routine wizard in chapter 2 (of course you can also use the manually configured check routine from section 2.5, **Camera Images in a Check Routine**, if you want to work from a camera image, but the remainder of this section refers to the check routine created using the check routine wizard to ensure identical working conditions).

The following dialog box shows all checks contained in the selected check routine with their titles and additional explanations. Entering meaningful texts in the **Check Properties** dialog will considerably ease the identification of existing solutions for similar image processing problems.

Select the first check “First bar code” by clicking the description text in the left column and leave the dialog with **OK**. The check is appended to the current check routine.

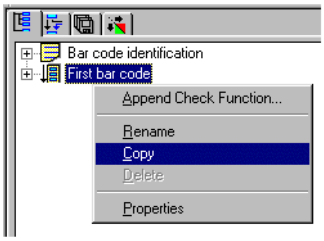
❷ 

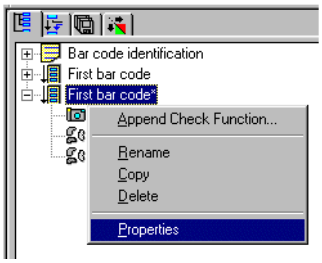
The character recognition will use the same image section as the bar code identification. Therefore the easiest way will be simply to copy the check just imported by choosing **Copy Check** from the **Edit** menu or the context menu of the check. Enter a corresponding description for the copied check, either by in-place editing or by choosing **Properties** from its context menu or the **Edit** menu.

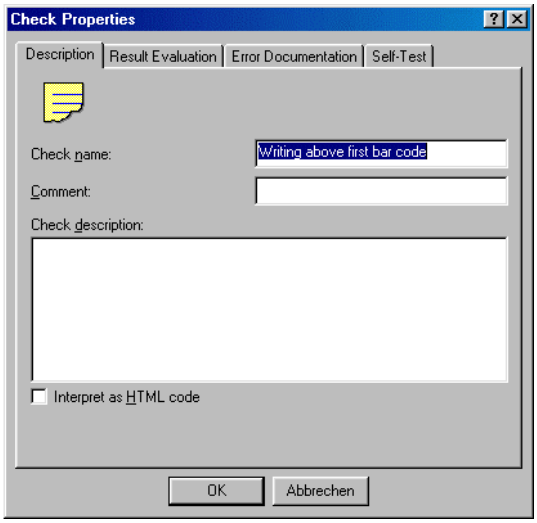


❶ 

Create a new check routine and import the bar code check from the check routine created in section 2.1







❷ 

Copy the imported check and enter a corresponding description for the copied check.

❸  

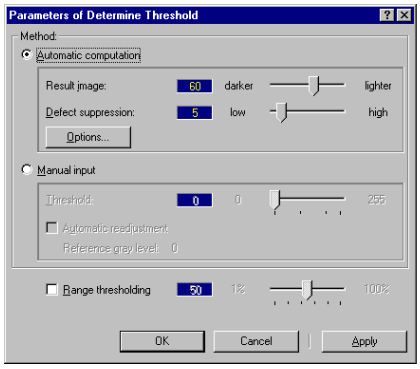
Now open the parameter dialog of function Define ROIs and adjust the region in **Select** mode. The rectangle should not extend over the bar code, because we do not want to find any of its lines as objects. The rectangle should be slightly lower than the characters (object search is done left to right, top to bottom. If one character is lower than the others, it will be found later and require a sorting operation according to x coordinates. Defining the region of interest lower than the characters will ensure that they are all found on the first search ray and be sorted automatically).Leave the parameter dialog with **OK** and remove check function Identify bar code by choosing **Delete** from its context menu.

❹   

Append function Determine threshold, Open its parameter dialog and adjust the brightness of the result image so that the characters appear unbroken in the thresholded image without fusing into each other or background artifacts. Append function Create ROIs by thresholding and execute it to see the characters as objects of their own.

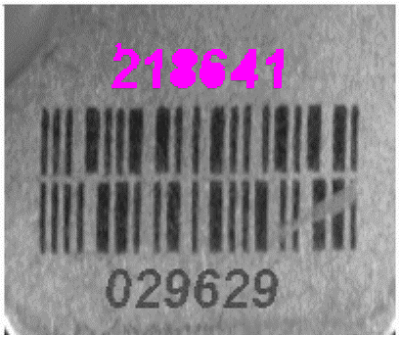
❸ 

Adjust the region of interest to cover the line of digits to slightly below the digits.



❹ 

Adjust the brightness of the result image thus that the characters are not interrupted but do not fuse with each other or with background artifacts., so that function Create ROIs by thresholding will be able to segment the characters as individual objects.



With this threshold, function Create ROIs by thresholding will be able to segment the characters as individual objects.

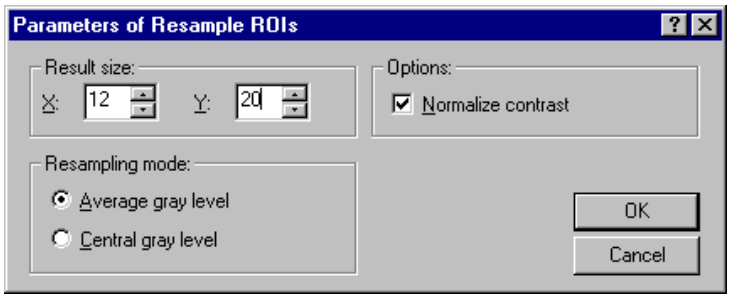
❺ 

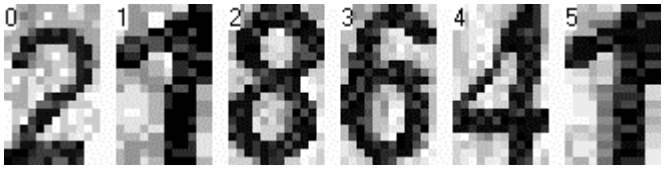
Classifiers used in character recognition need input data giving a good representation of the characters to be recognized. NeuroCheck uses resampled images of the characters which are normalized with respect to size and contrast. To create this representation append function Resample ROIs from the **Analysis** page and open its parameter dialog. At least 100 pixels are required to identify digits, but a higher resolution will improve recognition certainty. You should therefore increase the default setting of 112 pixels in an 8*14 grid to a resolution of 12 pixels in x direction and 20 pixels in y direction, in keeping with the typical width to height ratio of printed characters.

Execute the function. NeuroCheck displays the resampled images. You will notice that all digits are now of the same width and height. Furthermore they all make use of the full gray level range from total black to total white, which can be very important if illumination or background are not always identical. Thus the classifier will always receive input data with the same contrast properties.

❻ 

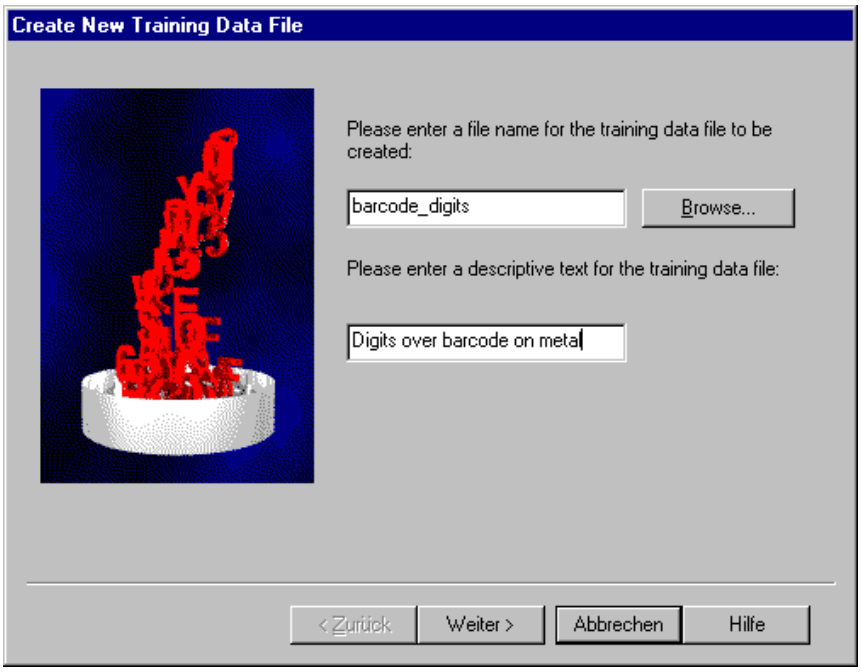
Append function Classify ROIs from the **Analysis** page. To fully configure this function a trained classifier is needed. Open the parameter dialog of the function and choose **Edit** in the **Training data** area. This will open the **Training Data Wizard** with **Create new training data file** as the only available option. Choose **Next** and enter a file name and a description on the second page (the file will be created in the directory set in the **System Options: Directories** property sheet for check routine data; alternatively you can use the **Browse** button to select directory and file name from a file selection dialog).





❺ Set the size of the resampling grid to 12 * 20 pixels to give the classifier some more input data. Note the check box **Normalize contrast**, which ensures that all images use the full gray level range.

NeuroCheck displays all digits with the same size and contrast.

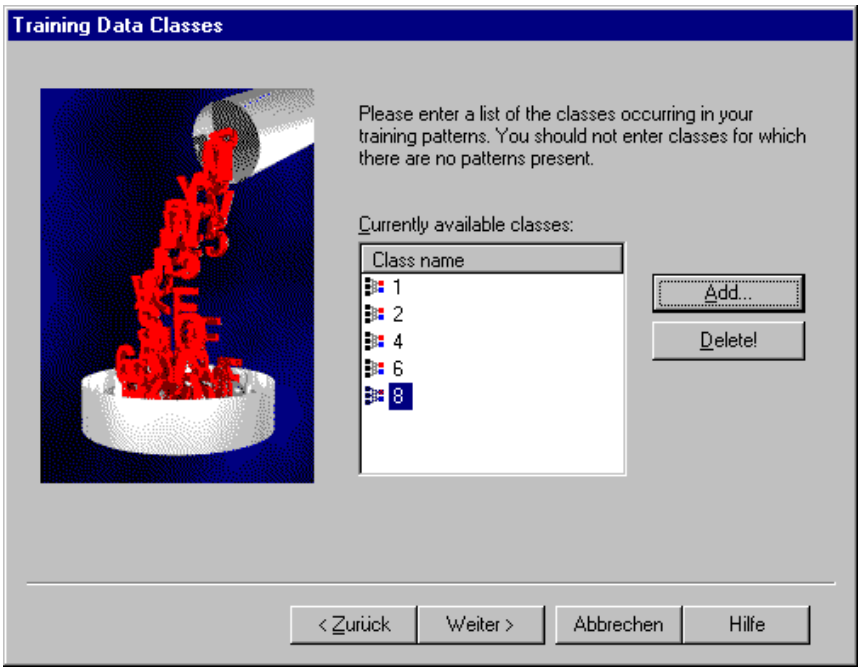


❻ In the parameter dialog of function Classify ROIs choose **Edit** in the **Training data** area to open the **Training Data Wizard**. Choose **Next** on the first page and enter a file name and description on the second page.

❼ On the following page you have to define classes for your patterns. We will now enter exactly the required classes. Of course you can add and delete classes later. Choose **Add**, enter `1` as the name of the first class, and confirm with **OK**. Proceed in the same way for classes `2`, `4`, `6` and `8`, i.e. exactly the digits present in the current set. You will notice that the **Next** button did not become available until at least two classes were present. This reflects the fact that the type of neural network classifier used by NeuroCheck is used to distinguish between classes. Therefore it needs at least two classes for training.

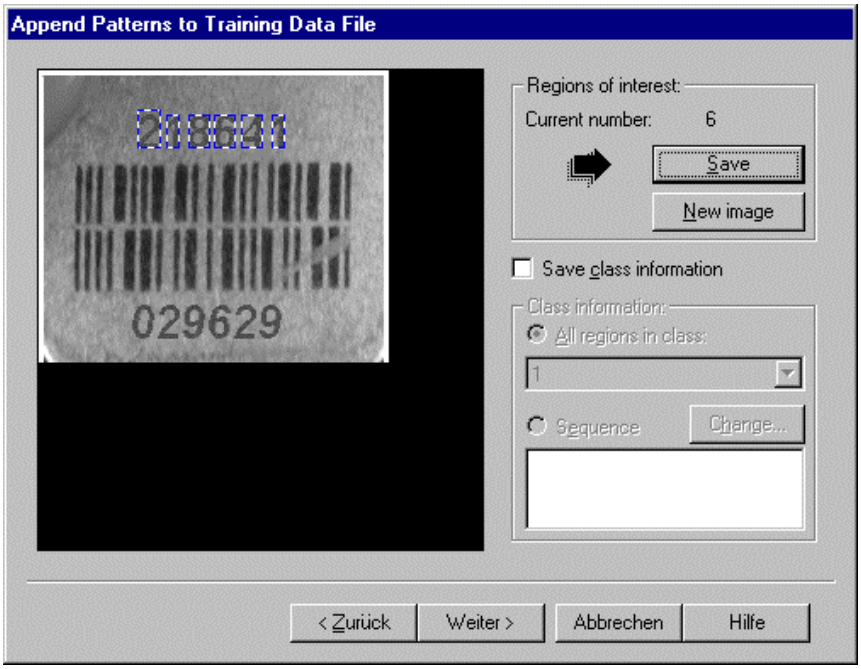
❽ Choose **Next** to switch to the **Append Patterns to Training Data File** page. The graphics panel of this dialog box displays the currently existing objects, i.e. the six digits. Choose **Save** to store the six patterns in the newly created training data file.

A real-world application requires pattern data from more than one image for training the classifier in order to compensate variations in print and image quality. The **New image** button allows you to execute the check in the background, including capture of a new camera image, to update the information on the page. That means, if you put a new test piece under the camera and choose **New image**, NeuroCheck executes the check in the background and presents the digits from the new test piece on this page so that you can store them in the training data file immediately. Thus you can very easily and rapidly create a sufficient collection of training patterns.



❼

Use the **Add** button to define the classes required for the training patterns.



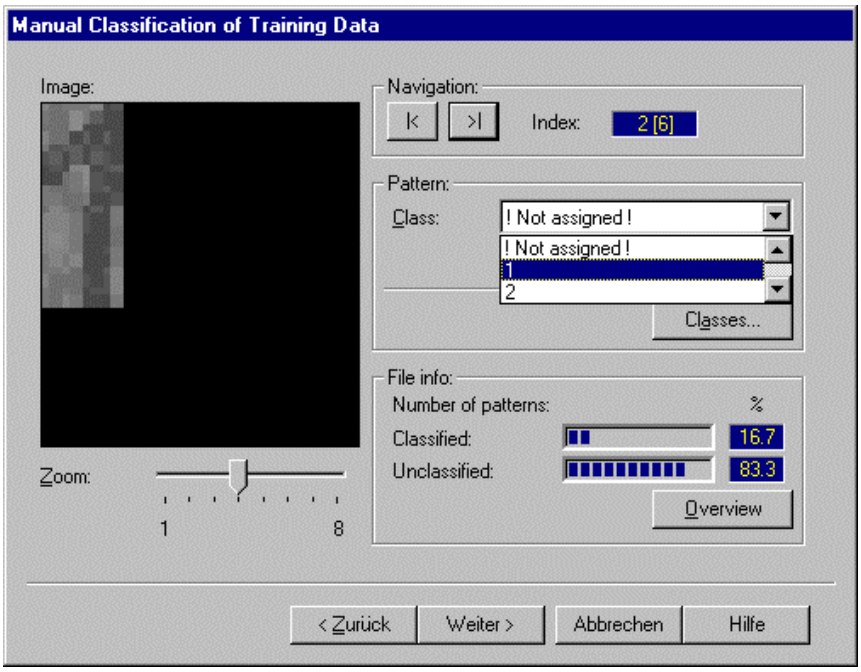
❽

Choose **Save** to store the patterns in the training data file.

You can create a sufficient collection of training patterns without leaving this page by using the **New image** button.

❾ On the last page of the **Training Data Wizard** you have to assign the correct class to each of the training patterns so that the classifier can learn to distinguish between the classes. Switch between the patterns using the navigation buttons and select the correct class for each pattern from the **Class** list box. It is always a good idea to check all patterns again after classification is finished, because classifiers can be very sensitive to errors in their training data.

After choosing **Next** you are back in the parameter dialog of function Classify ROIs.



❾

Assign the corresponding class to each of the patterns, using the navigation buttons to switch between the patterns. Choose the correct class out of the **Class** list box.

7.2 Creating a Classifier

Problem

You want to use the newly created digits to train a classifier.

Result

You know how to create and train a classifier.

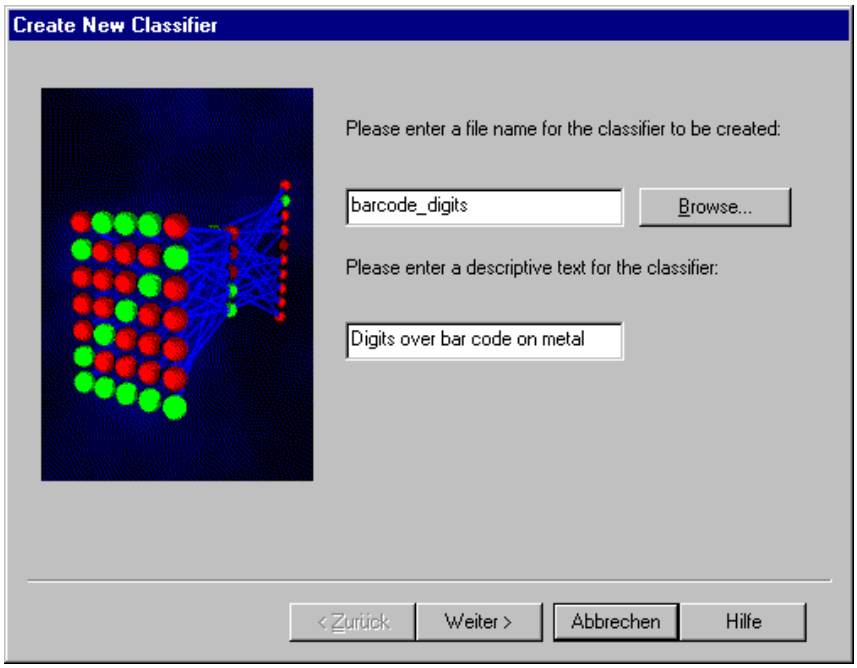
Solution

❶ Choose **Edit** in the **Classifier** area of the parameter dialog of function Classify ROIs. The **Classifier Wizard** appears with **Create new classifier** as the only option. Choose **Next** and enter a file name and a description on the second page. Training data and classifier file differ in extension, so you can use the same name as for the training data file to be able to identify the files belonging together.

❷ Choose **Next** to reach the **Training Data and Feature Selection** page. Choose **Browse** and select from the file selection dialog the newly created training data file. A list of features available for classification appears, which in this case only contains the **Resampled image** feature. Activate this feature.

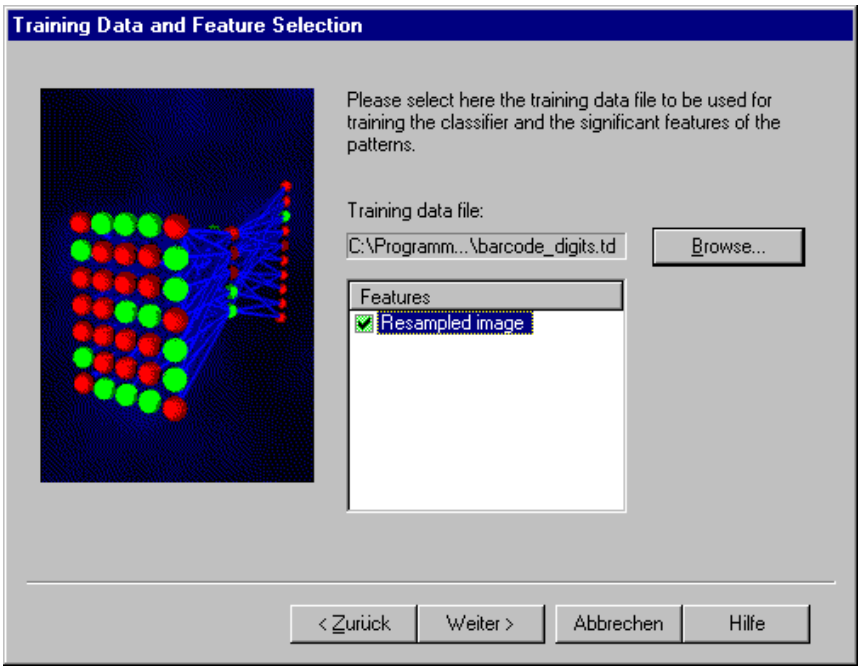
The purpose of this list is to exclude certain features from use by the classifier. For example, the **Origin X** feature may have been computed for the regions in order to sort them by x coordinates. It would not be meaningful to use this in classification, because the classifier would simply try to learn the sequence by heart during training, instead of considering the appearance of the patterns.

One more remark: it is possible to store patterns with different feature sets in a training data file. This allows you to add features later that are needed for some purpose, e.g. sorting, as described above. But the classifier requires all patterns to have the same set of features, i.e. it will use only features common to all patterns in the file. No feature added to the check routine after creation of the training data file will be available to the classifier - unless the training data file is recreated with these features or all patterns are deleted, which do not possess these features.



❶

Enter a file name and a description for the classifier on the second page of the **Classifier Wizard**.



❷

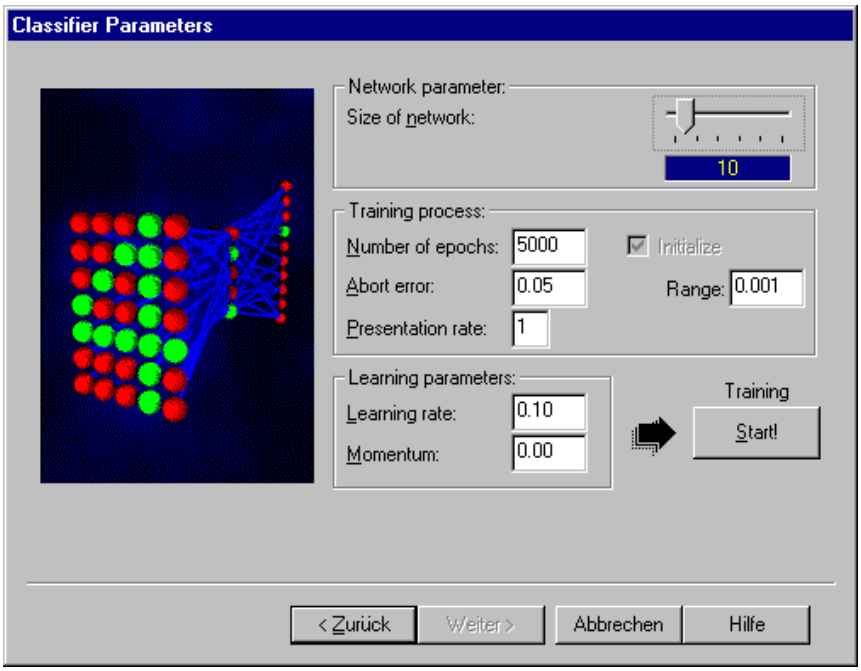
Select the newly created training data file using the **Browse** button and activate the **Resampled image** feature.

❸ Choose **Next** to switch to the **Classifier and Training Parameters** page.

Set the network size to 10. This parameter is largely a matter of experience and experiment, there are no fixed rules for it. You will find some comments on this topic in the "Classification" section of the online help and the manual.

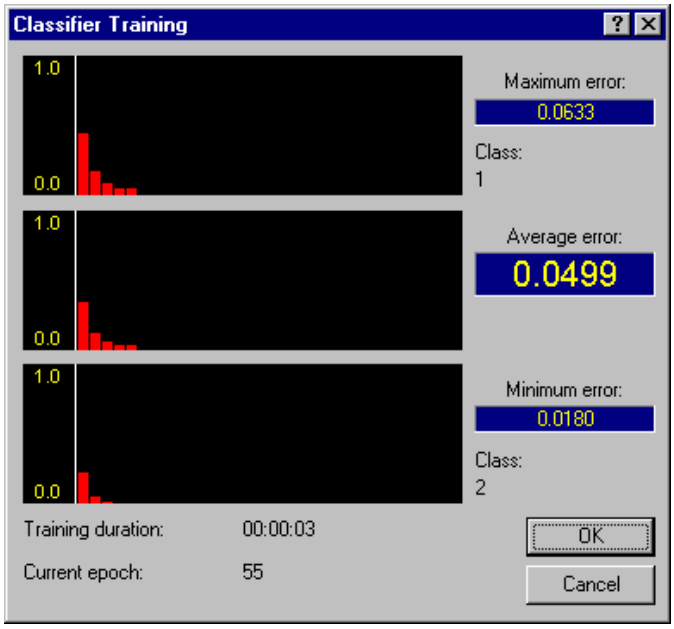
The default training parameters work well for a large class of problems, so you probably will not need to change them. Choose **Start** to start classifier training.

❹ The **Classifier Training** dialog visualizes the training process. After the prescribed maximum error has been reached, the **OK** button becomes available so that you can return to the **Classifier Wizard** to save the classifier by choosing **Next**. This will return you to the parameter dialog of function Classify ROIs to which the newly created classifier has been automatically attached. You can now leave this dialog with **OK** and execute the function.



❸

Set the network size to 10 and choose **Start** to start classifier training.



❹

After successful training you can return to the wizard by choosing **OK** and save the classifier by choosing **Next**.

7.3 Target Value Configuration

Problem

You want to use the newly created classifier to check the correspondence of bar code and plain text.

Result

You know how to use a classifier, how to set target values for the classification and transmit the classification result via serial interface.

Solution

❶ Close the parameter dialog of function Classify ROIs and append function Evaluate Classes from the **Analysis** page. Open its target value dialog by choosing **Target Values** from the **Check Function** menu or its context menu.

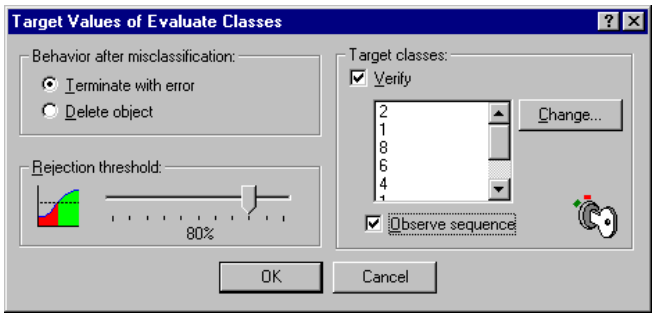
Set the **Rejection threshold** slider to 80%. This instructs NeuroCheck to accept only digits for which the classifier is more than 80% certain about the class to accept as correctly classified. Due to the fact that it is impossible to train a classifier on all possible patterns (a 12*20 resampled image can show more combinations of 256 gray levels in its 240 pixels than there are molecules in the universe) classification is a statistical discipline. Each classification result is accompanied by a statistical certainty computed from characteristics of the classifiers behavior, based on the observation that classifiers practically never err in such a way that they are completely sure about a wrong class.

Activate the **Verify** check box. This instructs NeuroCheck not only to classify the digits but also to compare the recognized classes to the target values. Activate the **Observe sequence** check box to tell NeuroCheck to accept only lines containing the selected digits in exactly the same sequence.

❷ Choose **Change** to open the **Edit Class Sequence** dialog in order to enter the requested sequence of digits. From the **Class selection** list select `2` as the class of the first digit and choose **Add**. Proceed in the same way for the following digits of the sequence `218641`. If you inadvertently select a wrong class, you can remove it from the **Class sequence** list by selecting it there and choosing **Remove**.

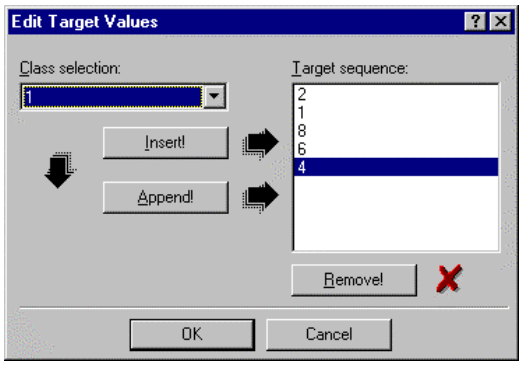
When you have assembled the class sequence completely, confirm with **OK**. You will be returned to the **Target Values** dialog of function Evaluate Classes, which will now display the class sequence in the **Target class** area.

❸ When you execute the check routine up to the last function now, NeuroCheck displays all recognized digits together with the respective recognition certainty. This certainty should be higher than 0.9, but values very close to 1.0 may indicate an overfitted classifier, which will not be very tolerant of image degradations.



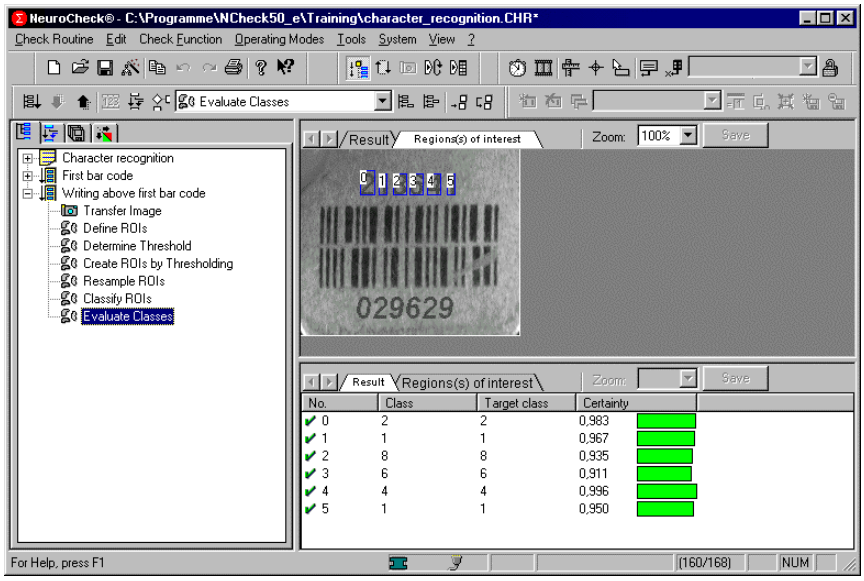
❶ 

Add function Evaluate Classes and open its target value dialog (shown here after the editing of the class sequence in the next step).



❷

Choose **Change** to open the **Edit Target Values** dialog in order to assemble the sequence of digits.



❸ 

NeuroCheck displays all recognized digits together with the respective recognition certainty.

❸ 

Switch to the **Output** tab page of the edit pane and activate the check boxes **Serial interface** in the global branch and the branches of functions Identify Bar Code in the first and Evaluate Classes in the second check.

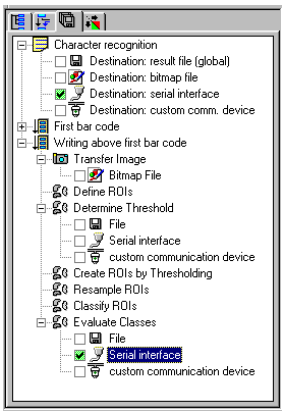
❹ 

Switch to automatic screen configuration mode. Open the parameter dialog of the existing check function visualization window by choosing **Parameters** from the context menu and activate the checkbox of function Identify Bar Code from the first check. From the **Display** list select **Result**.

Now create another check function visualization window by clicking the corresponding button in the screen configuration toolbar and drawing the window on the screen. Open the parameter dialog and activate the output of function Evaluate Classes from the second check. From the **Display** list select **Result**.

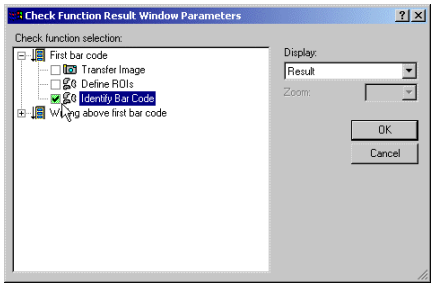
❺ 

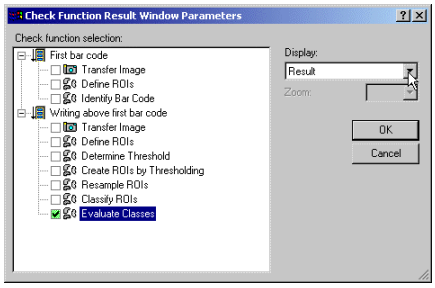
Switch to automatic mode and press the **Return** key. The check function visualization windows now display the results of functions Identify Bar Code and Evaluate Classes. At the same time both strings are transmitted via serial interface to a controlling instance.



❸ 

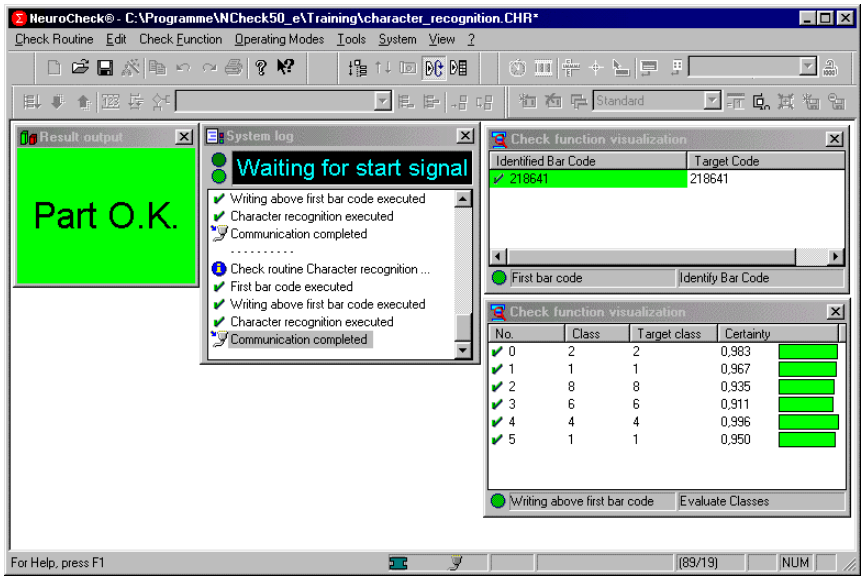
Activate the check boxes **Serial interface** in the global branch and the functions Identify Bar Code and Evaluate Classes.





❹ 

Activate the visualization of both check functions



❺ 

Switch to automatic mode and press the **Return** key. The check function visualization windows now display the results of functions Identify Bar Code and Evaluate Classes.

❺ Now let us see how the Classifier behaves when confronted with unknown data. Switch to manual mode, open the parameter dialog of function Define ROIs and move the rectangular region down to the line of digits below the bar code.

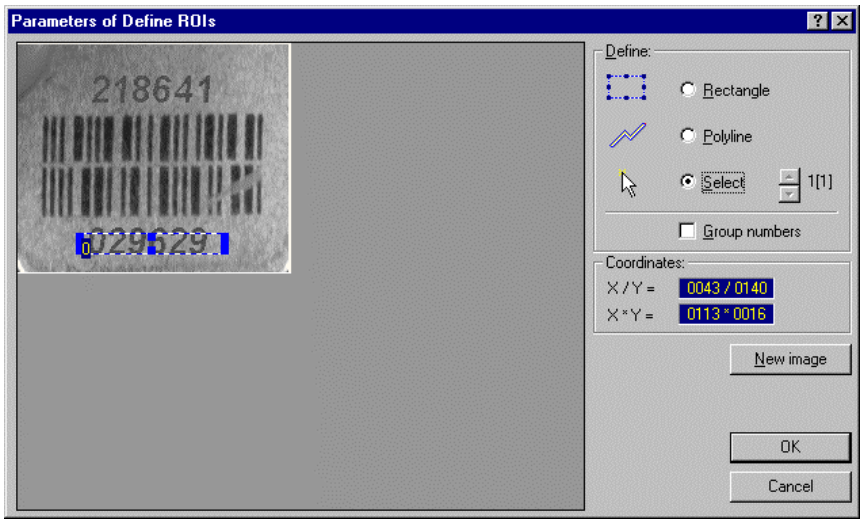
❻ Execute the check routine step by step. You will notice that everything works well, until you reach function Classify ROIs. The classifier will report errors, because the new sequence of objects contains digits it has never seen before.

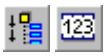
It does recognize the three digits belonging to known classes, namely the two 2's and the single 6, although the recognition certainty is lower than before. A closer look reveals considerable differences in the homogeneity and contrast of the digits. Keep in mind that the classifier has been trained with only very few samples for each class, so even small differences in the appearance of the patterns are able to cause this uncertainty.

The classifier also assigned classes to the three unknown digits, but with a much lesser certainty. This illustrates the reason why at least two classes are needed. Classifiers of this type will always assign a class, so applications with only a single class are not meaningful, because this class would always be assigned.

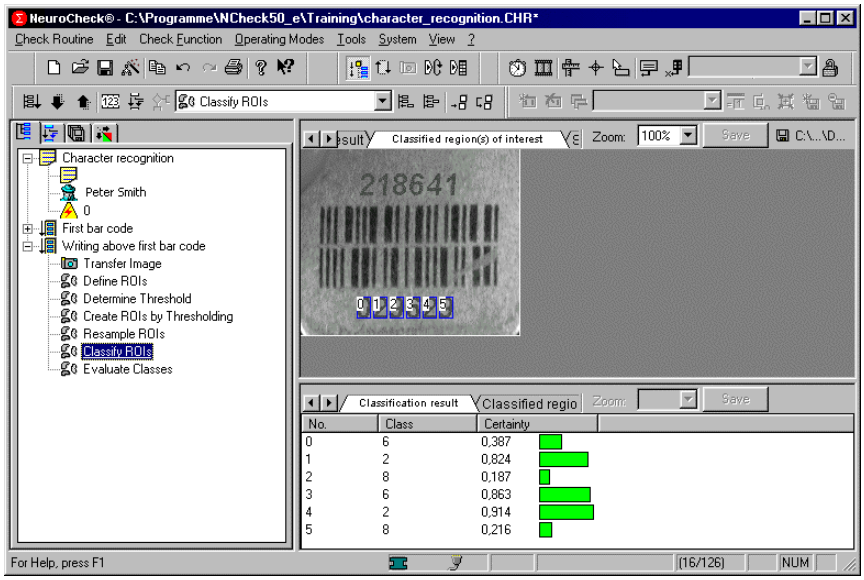
The classifier "recognizes" the two 9s as 8s, the 0 as a 6, but the certainty falls far short of the rejection threshold (defined in the next function Evaluate Classes), so no harm will be done.

Please note that the actual numerical certainty results may differ between different classifiers because due to the random initialization, two classifiers are never precisely identical, although they will perform equally well after training.



❺ 

Move the region of interest to the bottom line of digits.



❻ 

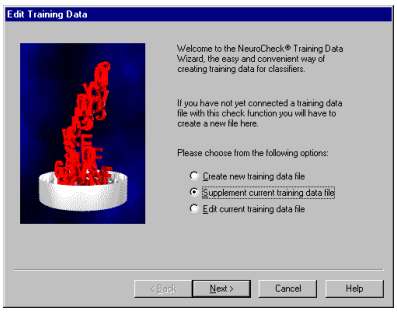
The classifier correctly recognizes the known digits and assigns a low certainty value to the other objects.

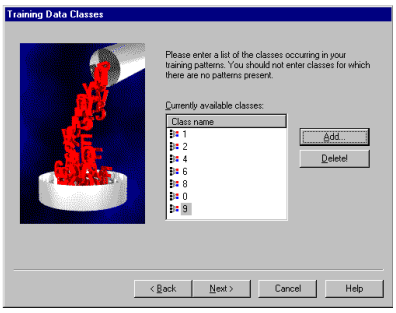
❻ In order to update the classifier to recognize the new digits, open the parameter dialog of function Classify ROIs, choose **Edit** from the **Training data** area. The **Training Data Wizard** appears with the correct option, **Supplement current training data file** preselected. Choose **Next**. On the **Training Data Classes** page choose **Add** and add the classes `0` and `9`.

❼ Choose **Next**. On the **Append Patterns to Training Data File** page choose **Save** to store the new patterns. Choose **Next** and assign the correct classes to the new patterns on the **Manual Classification** page. Then choose **Finish** to return to the parameter dialog of the function.

❽ Now choose **Edit** from the **Classifier** area. The **Classifier** appears with the correct option, **Edit current classifier** preselected. Choose **Next**. On the **Classifier and Training Parameters** page choose **Start**. If the classifier fails to learn the new patterns it may be adapted too fixedly on its old training data set. In that case choose **Cancel** in the **Classifier Training** dialog (**OK** will not be available for an untrained classifier) and activate the **Initialize** check box in the wizard. NeuroCheck will then create a completely new classifier, using all the patterns in the data set instead of trying to update the old classifier.

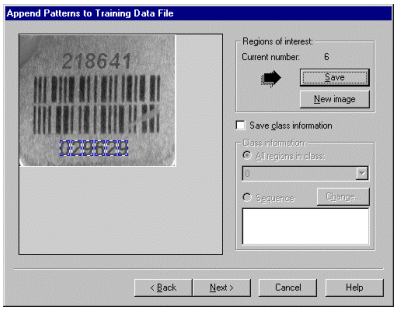
After successful training choose **Finish** on the last page of the **Classifier Wizard** and close the parameter dialog of function Classify ROIs with **OK**.

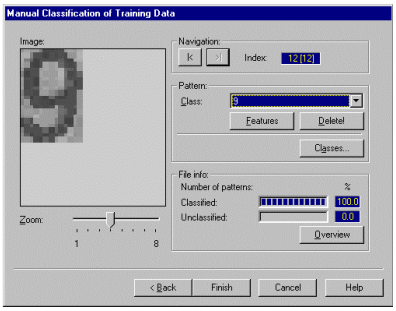




❻ 

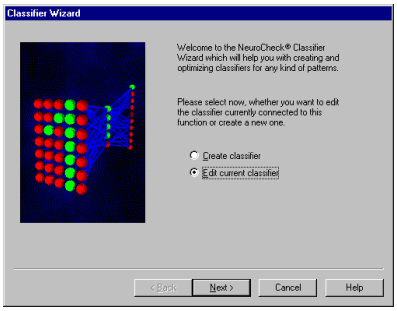
From the parameter dialog of function Classify ROIs open the **Training Data Wizard** to supplement new classes and patterns.

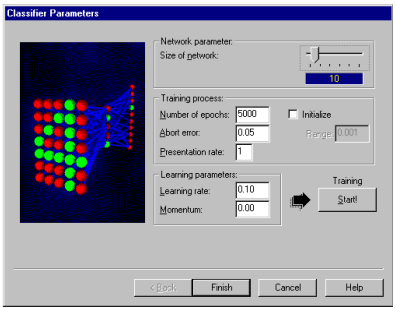




❼

Store the new patterns and classify them on the final page of the **Training Data Wizard**.



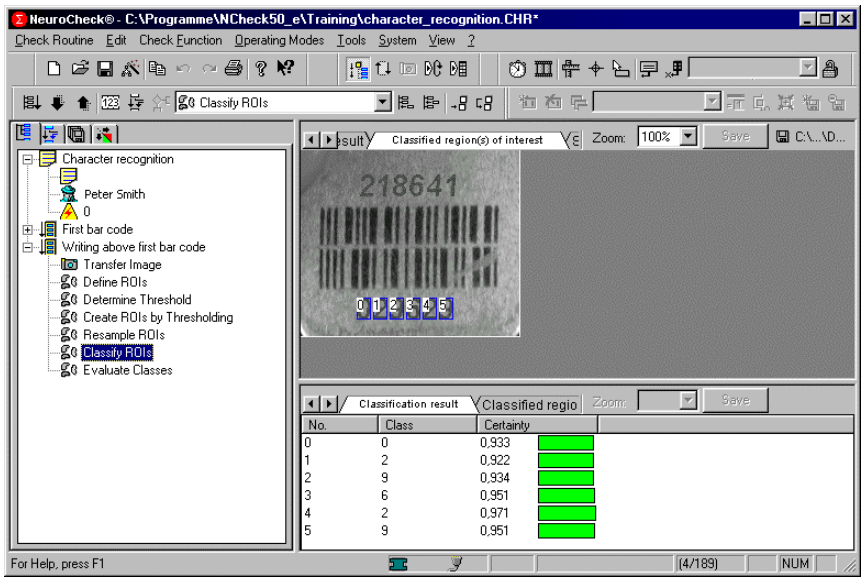


❽

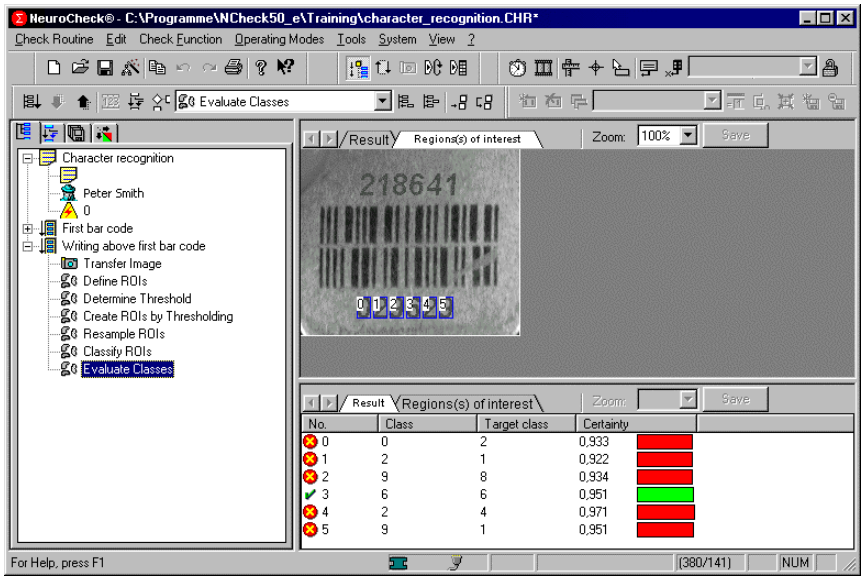
From the parameter dialog of function Classify ROIs open the **Classifier Wizard** to edit the current classifier by retraining it with the new patterns. Leave the wizard after successful training.

❾ Execute the check routine again up to function Classify ROIs. The function will now recognize all digits with sufficient certainty.

❿ Function Evaluate Classes will of course still report an error, because the sequence of classes does not correspond to the target sequence. Only the 6 in the fourth digit is still considered correct.



❾ The updated classifier recognizes all the digits.



❿ Function Evaluate Classes notes the incorrect sequence of the digits.

8 Template Matching

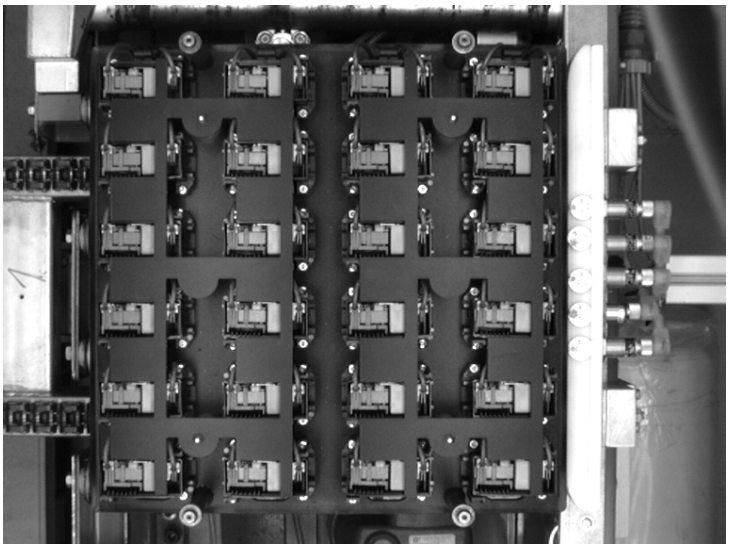
In the previous chapters the searching for relevant objects in the image was done by binary thresholding. The created ROIs were filtered using several features with Screen ROIs.
Under the following circumstances this method will fail

- if no clear contour determination of the object in each direction is possible,
- if a single object consists of multiple unconnected fragments,
- if objects merge with other objects using the binary threshold.

An alternative way to localize objects in the image is the matching of the image information with a previously defined template. Thereby the template will be compared with all possible image positions and for each of the positions a certainty value for the match will be calculated. The positions with the best results will be created as new ROIs. These ROIs could be used for e.g. a presence verification or a position adjustment.

In this chapter you will learn

- how to search for objects in images with template matching,
- how to optimize the templates.



Board with multiple similar elements

8.1 Searching Objects with Template Matching

Problem

You want to ascertain that all elements on a circuit board are present.

Result

You know how to use the function Template Matching for an object search in an image.

Solution

You have a circuit board on which nine elements should be placed. The check routine has to identify and locate these elements in a given image.

Create a new check routine. Enter “Template Matching“ as the name of the check routine and rename the first check with “Check elements”.

First append the check function Transfer Image and open its parameter dialog. Select the **Browse** button and choose `board.bmp` from the `training`-directory as your image-file.

Append Define ROIs to the first check. Define a rectangle area and resize it to “Full Image“ using the context menu.

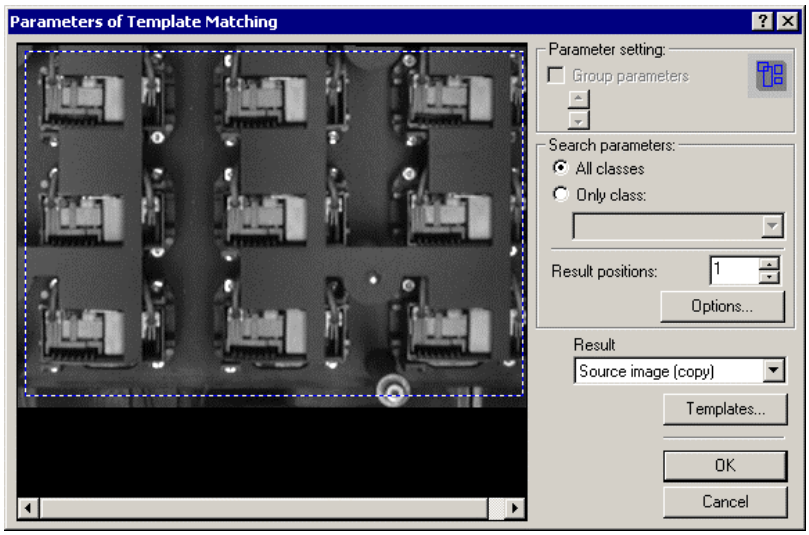
❶ Now add the check function Template Matching from the category Objects.

Open the parameter dialog of this function and choose **Templates** to start the **Template Wizard.** If you have not stored a template before, you can only choose the option **Add templates** on the first page. Choose **Next** to proceed with the second page.

❷ In this dialog it is possible to define multiple template classes for this check function. In our example we only need one class.

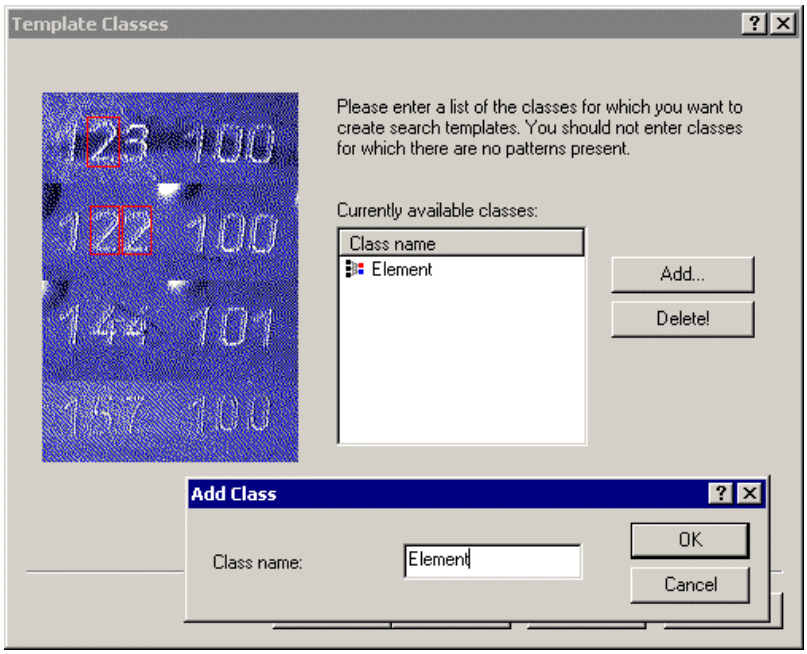
With **Add** you open a dialog in which you can enter the name of the new class. Enter the name “Element“ and confirm with **OK**.

Choose **Next** to go to the next page of the **Template Wizard**.



❶ 

Use the **Templates** button in the parameter dialog of Template Matching to open the **Template Wizard**.



❷

Add a new class named “Element”.

❸ On this page you can collect the search patterns.

Adjust the size of the rectangle area for the template to 70 pixel width and 45 pixel height. (Remark: It is possible to define areas with different dimensions for other templates.)

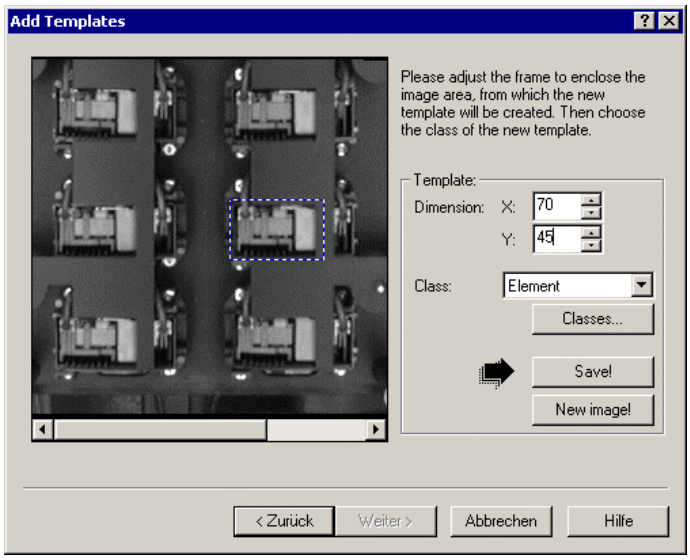
Move the rectangle by pressing and holding the left mouse button so that it encloses Place the image cursor over one complete element via drag and drop.

Choose **Save** to store the contents of the area as pattern for the specified class.

Don't forget to press the **Save** button to copy the template into the properties of the check function.

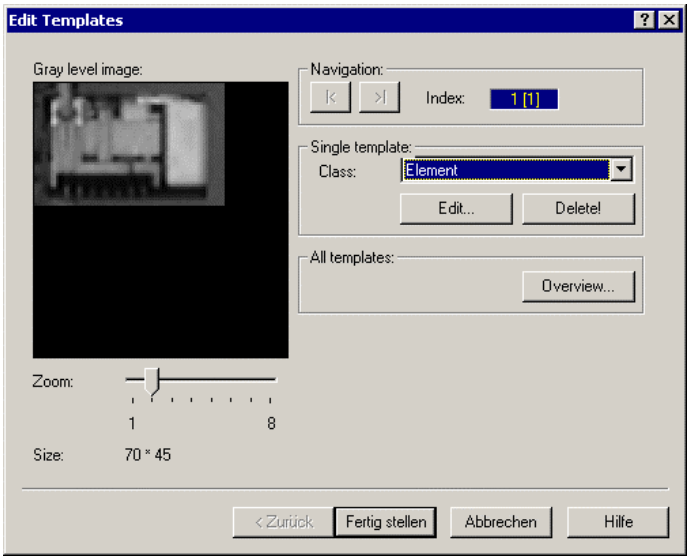
❹ Choose **Next** to go to the final page. Change the zoom factor for the image to 2.

For now there are no further changes of any settings in this dialog, so leave the template definition with **Finish**. After that you are back in the parameter dialog of the check function Template Matching.



❸

Define a rectangular area (of 70 x 45 pixel in this example) that encloses the search pattern. Confirm with **Save** to store the template.



❹

You can see the assigned class name for the template.

❺ We are searching for nine elements of the same type. You can enter the number of templates to search for in the field **Result positions**. To demonstrate the functionality of the template matching enter the value `10` in this field.

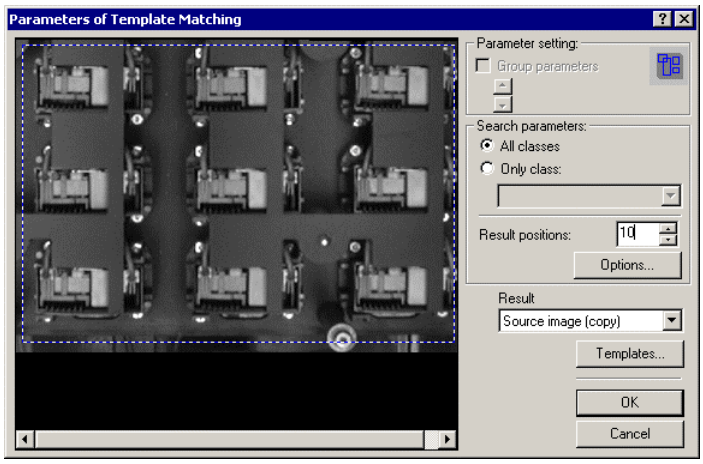
The check function searches for ten objects in the image which have the best match results with the previously defined reference template.

Leave this dialog with **OK** and execute the function.

❻ NeuroCheck displays the found objects in the result view. Only nine templates corresponds to the search criteria and are displayed.

Split the result window, so that you can see the class names and the certainty values in the first window and the found positions in the image in the other window. The certainty represents the correlation between the reference template and the pattern found in the image. This value isn't exactly 1.0, even at the position of the previously defined template, because template matching uses internal optimizations in order to enhance the performance.

Now you could use the check function Count ROIs to evaluate the correct number of elements on the board.



❺

Enter the value `10` in the field **Result positions** to search for 10 objects matching the defined template.

❻ 

Execute the check function and split the result window to see the detected regions and the classification results for the found objects.

8.2 Optimization of Template Matching

Problem

You want to increase the performance of the template search.

Result

You know how to decrease the number of grid points, used for the template matching. Further you know other options of the check function Template Matching.

Solution

In general the execution time of the check function increases linear with the number of templates. If you need to store more than one reference template to ensure a correct detection of an object, you thus increase the time consumed by the function.

To optimize the performance of Template Matching there are several possibilities in NeuroCheck.

❶ Copy the last check and rename it to “Check elements (optimized)“.

Open the parameter dialog of Template Matching and start the Template Wizard with **Templates**. Press **Next** to proceed with editing the template.

❷ In order to edit the template choose the **Edit** button on page **Edit Templates**.

On the first page of the **Template Wizard** in **Edit Mode**, leave the option on “automatic“ and select **Next**.

❸ Enter a new value for the number of grid points (correlation points). The less grid points you use, the less execution time is consumed for the template matching. In this example we use 50 points.

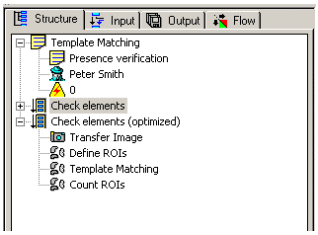
Continue to the last page by pressing **Next**.

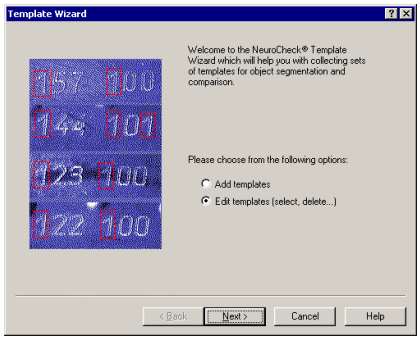
Now you can choose two methods for an automatic optimization of the grid points:

1. **Area-controlled**: The grid points will be distributed uniformly on lighter and darker areas of the template.
2. **Edge-controlled**: The points will be set in a specified distance (neighborhood) along the edges of the object.

The best configuration for these options depends on the size and structure of the object. In this case we are using the edge-controlled method with a neighborhood area 5 pixel.

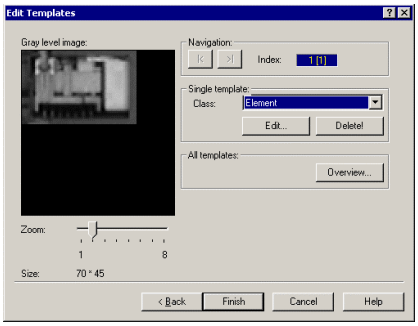
Press **Start** to perform the optimization then close this dialog with **Finish**.

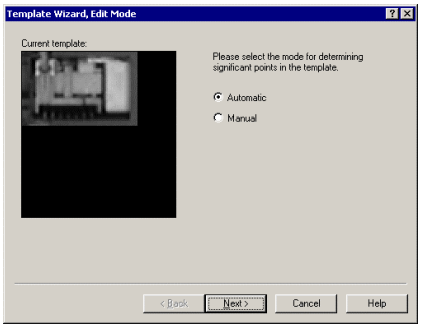




❶ 

Copy the check and rename it. Open the parameter dialog of Template Matching and start the **Template Wizard**. Choose **Edit templates** and proceed with **Next**.

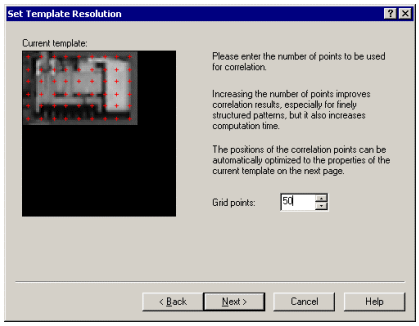


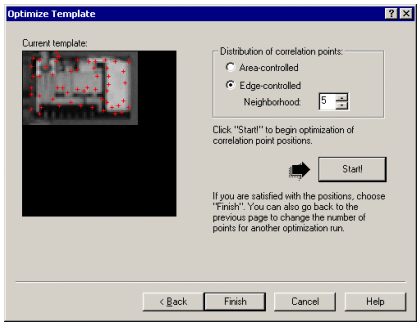


❷

Enter the edit mode with **Edit**.

Select **Automatic** and press **Next**.





❸

Change the number of grid points to 50.

In the following dialog the distribution of the points can be optimized by selecting a distribution method and pressing **Start**.

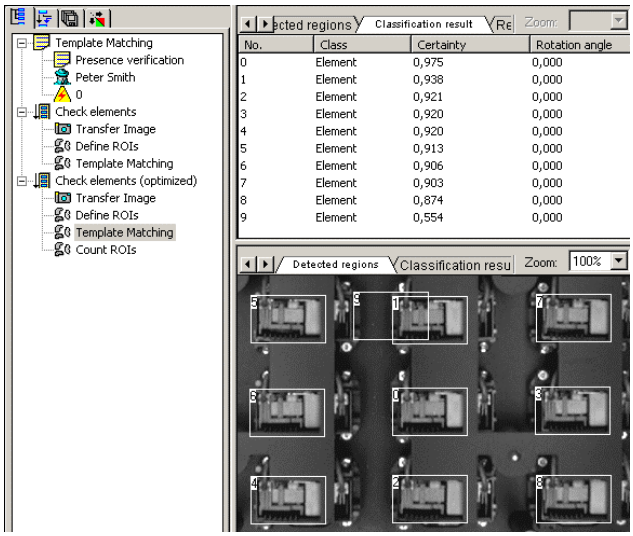
Back on the last page of the Wizard choose **Finish**.

Close the parameter dialog of the check function with **OK**.

❹ Execute once more the function Template Matching. This time 10 objects are detected with one invalid position which has a very low certainty. The performance of the template matching has increased, however, for this example you will hardly notice this on a fast computer (600 MHz and above).

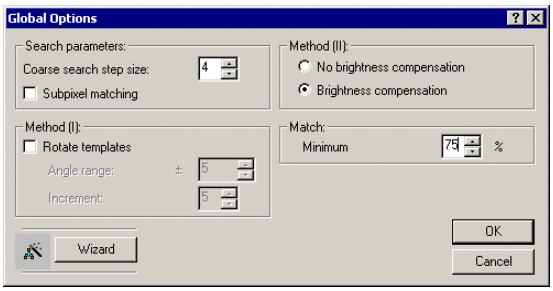
❺ The invalid object can be excluded using the certainty value. To do this, open the parameter dialog again an select **Options**. Set the value for **Minimum** for the option **Match** to 75 % and confirm with **OK**.

❻ If you execute again, you will see that only templates with a certainty above 0,75 are found.



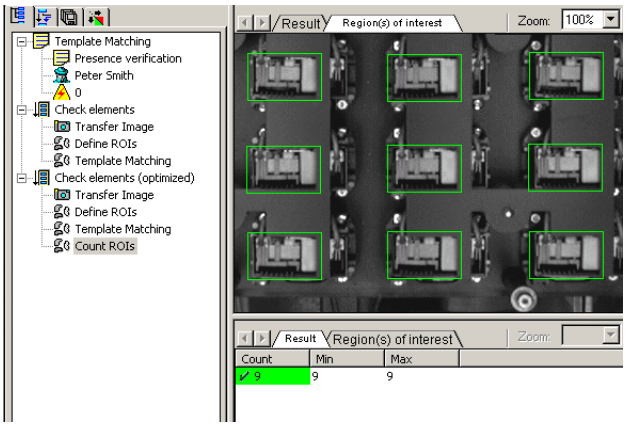
❹

Execute again the check function Template Matching. Now 10 templates are detected. In this example, an invalid object was found with a certainty of 0,554.



❺

Open the parameter dialog of Template Matching and set the **Minimum** for a **Match** to 75 %.

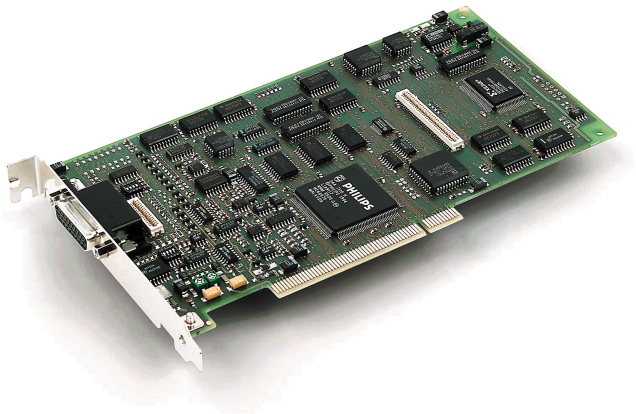


❻

Now only the correct nine templates will be found.

9 Advanced Topics

This chapter provides you with information concerning several advanced topics such as the analysis of error images, the configuration of image acquisition using a frame grabber, or the configuration of the serial interface.



Frame grabber board

9.1 Diagnostic Methods

When a check routine does not perform as planned, perhaps rejecting a test part in automatic mode as ‘not O.K.’ without obvious reasons, NeuroCheck offers various possibilities to determine the cause of the problem.
For example, you can configure a screen layout especially for debugging purposes. Or, in case of an ‘not O.K.’ signal, the images in question can be saved in a file for later analysis.

9.1.1 Configuration of a debugging screen layout

In automatic mode you can monitor the behavior of a check routine online. Additional debugging output may clutter the screen for normal operation. Since a check routine can manage any number of different screen layouts, you can create a special layout for debugging purposes. But a carefully designed screen layout can also be used during normal operation and thus provide valuable diagnostic information all the time.

❶  
Switch to automatic screen configuration mode. From the **View** menu choose **Screen Layout ▸ New**. Enter a description in the **Add Screen Layout** dialog and activate the **Use template** check box. From the list of templates select **Debug**. A new screen layout will appear, containing a status output window each for digital I/O and serial communication, one for execution time analysis, a system log, and a check function visualization window.

❷ To receive more detailed information concerning the type of failure of the check, it is possible to define custom status messages for each check or check function. There are three different possibilities:

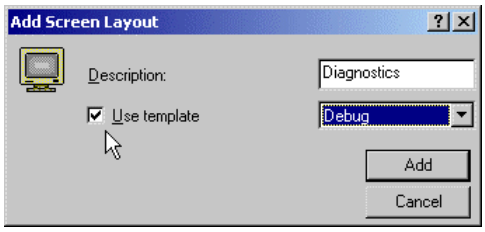
1.) Individual check / check routine yields ‘O.K’.

2.) Individual check / check routine yields ‘not O.K.’ because a check function couldn’t be executed properly.

3.) Individual check / check routine yields ‘not O.K.’ because of a target value violation, e.g. because a value is outside the allowed tolerance range.

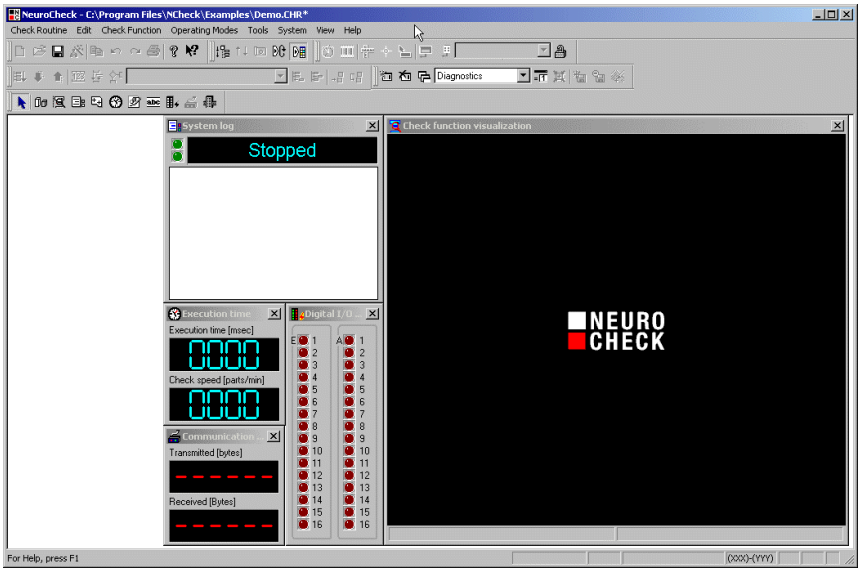
❸ 
To display a status message, the bottom area of a check function visualization window can be used. For this, activate **Display name and result state of check function** in the **System ▸ Settings for Automatic Mode ▸ Display** menu.

You can also use an additional window for a message. For this, create a result output window and open the parameter dialog. Select the category **Status Message**. Click **Continue** and select the check / check function you want to examine and confirm by clicking **Finish**.



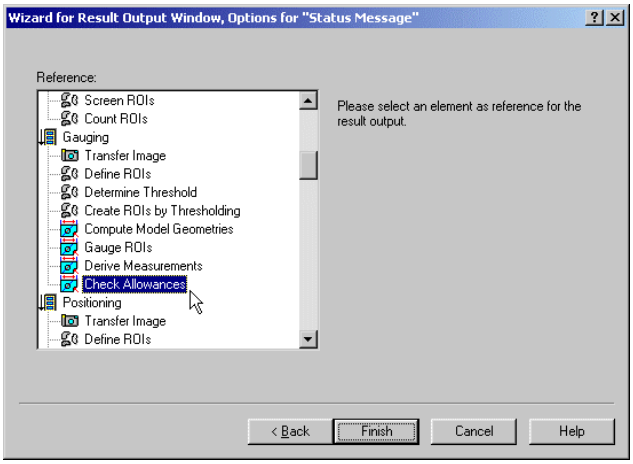
❶ 

Create a screen layout based on the **Debug** template.



❷

The new layout contains: system log, digital I/O status, check function visualization, communication status, and an execution time window.



❸ 

Select the check function that you want to generate a status message.

❹ 

The texts for the messages are defined in manual mode. Switch to manual mode and select the element (check routine, single check or check function) for which you want to change the message. On the Result Output Page in the Properties dialog, enter the text for each of the three possibilities.

Furthermore, you can change the output color from red to yellow or magenta in case of a target value violation. This setting is changed in the
System ▸ General Options ▸ Colors ▸ Signal color for target value failure menu**.**

9.1.2 Activate saving of error images

❺ 

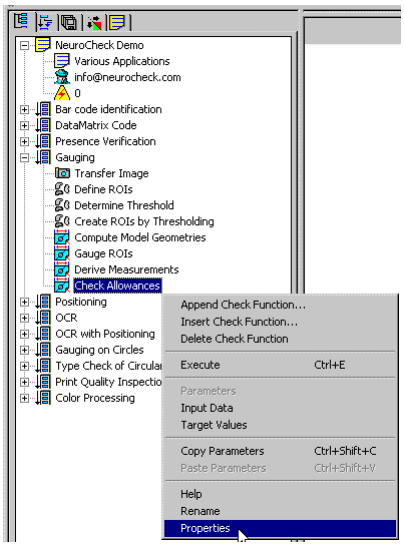
You can monitor error causes online individually for each check in a check routine by saving error images. In manual mode select a check and choose **Properties** from its context menu or the **Edit** menu. Switch to the **Error Images** page in the **Check Properties** dialog. Here you can activate automatic storage of error images by activating the **Save image** check box. Whenever this check terminates with 'not O.K.' it will now store an image documenting this error. The check function generating the image to be stored must be selected in the list below the check box. Typically the function Transfer Image will be used because everything else can be reconstructed from the camera image.

NeuroCheck stores files with the same base file name, numbered from 000 to 999. In the example the base file name is `bci` for bar code identification. The first file will be `bci000.bmp`, the last file `bci099.bmp` if 99 was entered as the maximum index. After file `bci099.bmp` has been stored, file `bci000.bmp` will be overwritten. The directory can be chosen using the **Browse** button.

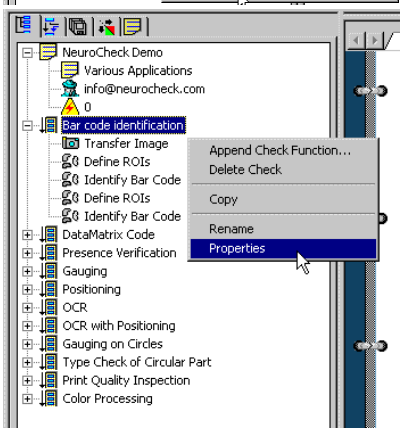
Activating the **Date/time stamp** option causes NeuroCheck to overlay the date and time, when the image was generated, over the bottom right corner of the image. Normally, this option should not be used as it causes the original image to be changed. The time of saving can also be found out from the file system using the Windows Explorer.

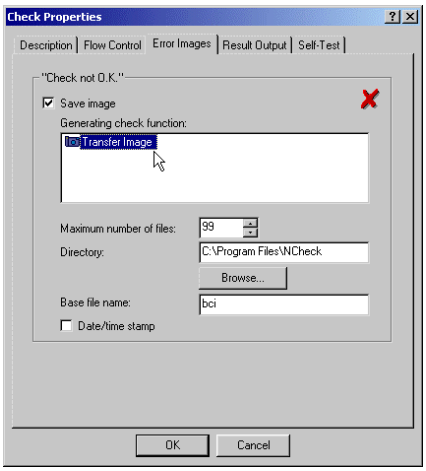
Note that error image storage works only with camera images, because it would not make much sense storing the same image over and over, when NeuroCheck computes a 'not O.K.' result from a bitmap. Also this option will work in automatic operation only to allow the user to control image storage in manual mode himself.

These error images can then be evaluated offline (perhaps after copying them to a different computer) step by step by opening the bitmap files in Transfer image. If you do an evaluation in automatic mode, do not forget to deactivate the saving of error images. Otherwise the original error images you are analyzing could be overwritten.



❹ To define individual status messages, change the result output for the check function in the **Properties** dialog.





❺ NeuroCheck can automatically store an error image for each check in the check routine.

9.2 Troubleshooting

In case that the error does not result from the check routine but from the hardware used or some other erroneous behavior of NeuroCheck, you can send an error message to NeuroCheck GmbH or one of our partners.

9.2.1 Software Trouble Report

If you send such a message, we need certain information. Among the information needed is the NeuroCheck version you use (including service pack number), your operating system and hardware status. To guarantee fast processing, please use the software trouble report.

❶ Use the **Help ▸ Feedback ▸ Bug Report** menu command to create a text file containing a form with system information. This file is opened immediately so you can edit it. Enter your name and your message. You can send this file as an attachment of an email.

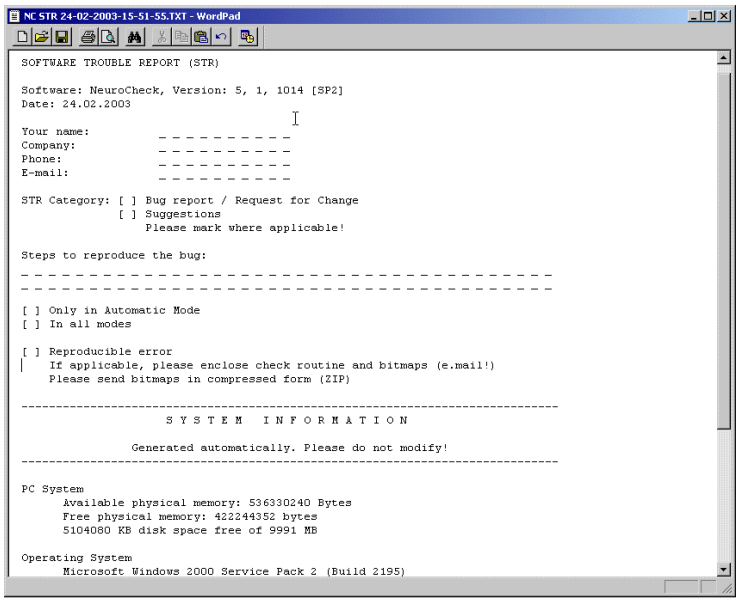
9.2.2 Log File

In certain cases it can be necessary to log NeuroCheck's behavior during execution. For this, active the creation of a log file.

❷ Choose **General Options** from the **System** menu to open the **General Software Settings** dialog box. On the **Debug** page you can activate system logging. If the **Create log file** check box is activated NeuroCheck writes information about its activities, e.g. name of the check routine, date and time of execution, check functions executed etc., to the log file mentioned before. The size of this log file can be limited for automatic inspection applications, which may run for days without interruption.

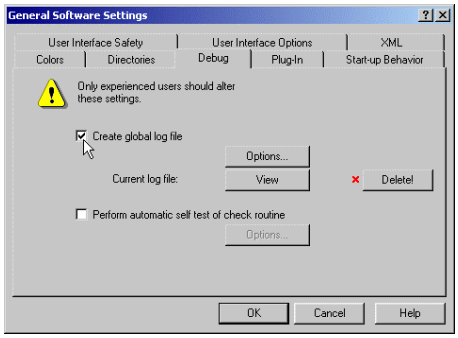
❸ The **View** button opens WordPad with the log file so that you can read the log file without having to leave NeuroCheck.

The status bar displays an additional symbol indicating that logging has been activated. Writing the log file needs time thus slowing down program execution. Therefore, deactivate the protocol option in day-to-day operation.



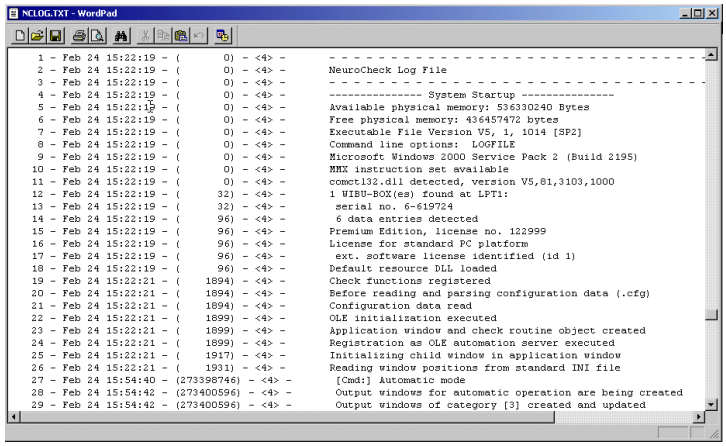
❶

A software trouble report is easily created. It contains important system information for troubleshooting.



❷

The **Debug** page of the **General Software Settings** dialog also allows you to create a log file recording NeuroCheck's actions.



❸

NeuroCheck records program execution in the log file.

9.3 Configuration of the Image Acquisition for frame grabbers

Problem

You want to install a frame grabber instead of a FireWire camera for capturing images with NeuroCheck.

Result

You are able to configure a frame grabber board for use with NeuroCheck.

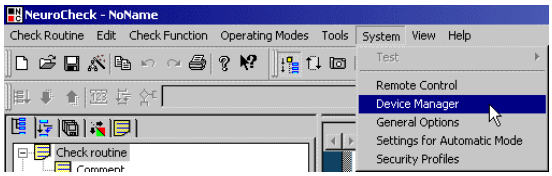
Solution

❶ After installing the hardware in your computer you have to set up the low lever driver delivered by the manufacturer of the board. Then you have to register the frame grabber board for the use in NeuroCheck. From the **System** menu choose **Device Manager**.

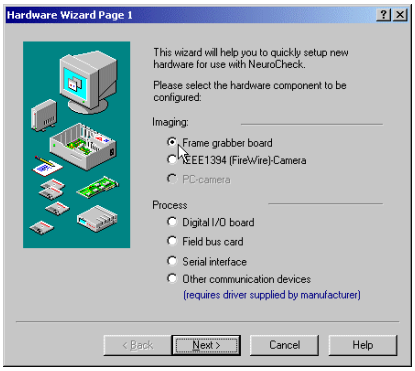
❷ Choose **New** in the **Device Manager** dialog box. The hardware wizard will guide you through the setup process. On the first page of the hardware wizard select **Frame grabber board**. On the second page select the board model.

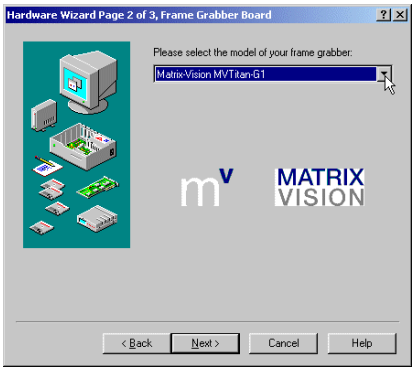
❸ As soon as you confirm your selection on the final page of the hardware wizard, the board is being tested. When the board has been found and its reactions verified, it is entered into the device tree. Note that the pertaining camera inputs are automatically added, too

❹ Normally you have to adapt the parameters of the used camera to the frame grabber it is connected to. But the configuration by using the properties dialog requires expert knowledge (in contrast to configuring FireWire cameras). If a camera setup file (`*.csu`) is available for your camera - frame grabber constellation you can load it by choosing **Import**.

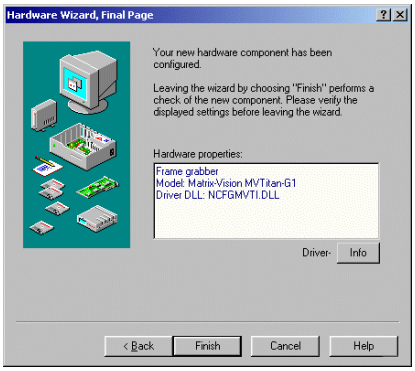


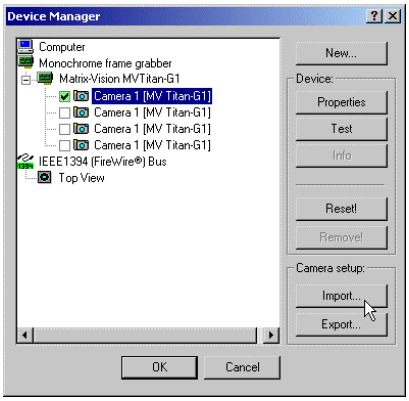
❶ From the **System** menu choose **Device Manager** to configure your frame grabber board



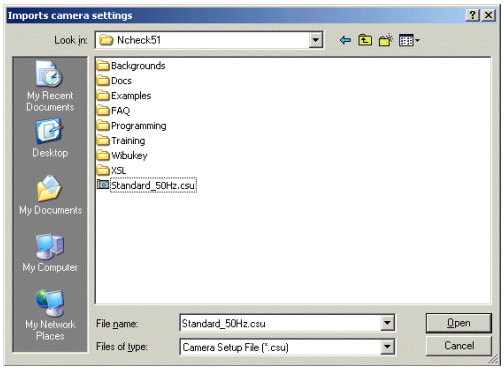


❷ After choosing **New** in the Device Manager dialog, the hardware wizard will guide you through the setup process.





❸ After confirming your selection the frame grabber is tested and added to the device list.



❹ Select the camera you want to configure in the device manager and choose **Import**. A file dialog is opened and you can load an existing `*.csu` file

9.4 Configuration of the Serial Interface

Problem

You want to establish communication between NeuroCheck and the control units of your manufacturing process (PLC/master computer) via serial interface.

Result

You are able to load, configure and test the device driver for serial communication.

Solution

❶ From the **System** choose **Device Manager**. In the **Device Manager** dialog choose **New**. On the first page of the **Hardware Wizard** select **Serial interface** and choose **Next**. On the second page you might have to change the setting for the port to be used. Default is COM2 Choose **Next** and confirm the final page with **Finish**.

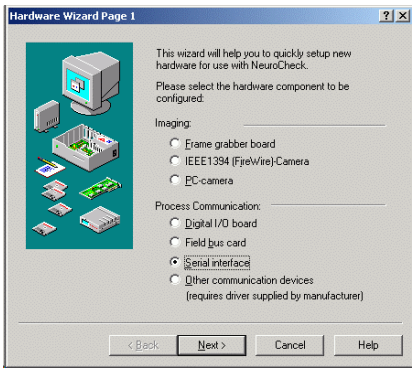
❷ Select the newly added serial interface entry in the **Device Manager** dialog and choose **Properties**. The configuration dialog of this device driver DLL offers the usual parameters for serial data communication.

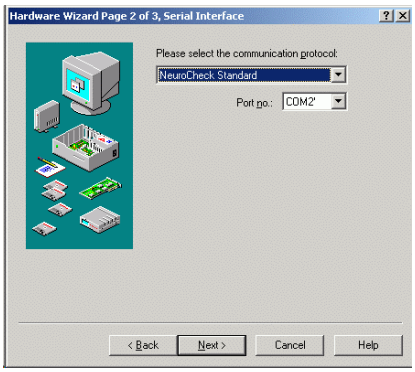
The port for serial communication has already been selected in the **Hardware Wizard**. The parameters available in this configuration dialog have to be adjusted to your process environment. Typical parameters are already set as default values: transfer speed 9600 Baud, eight data bits, one stop bit, no parity checking, flow control by XON/XOFF signals.

In the **Data packet format** area of the dialog box you should activate the check box **Append CR/LF** for communication with a terminal program. Every data packet is then send in a line of its own, improving readability. This setting will typically not be used for communication with a PLC. On the other hand, the check box **Append error check bytes** will typically not be activated for communication with a terminal program, whereas a PLC can use these signals to check the transferred data for errors. Leave the dialog box with **OK**.

❸ Now choose **Test** in the **Device Manager** to test communication via the serial interface. In the **Output data** field you can enter characters to be sent by NeuroCheck as decimal ASCII values. Separate the individual character codes by spaces. The **Send** button sends the characters over the serial interface to the terminal program. The **Input data** field displays the ASCII character codes most recently received by NeuroCheck from the serial interface. Thus you can easily check data transfer in both directions.

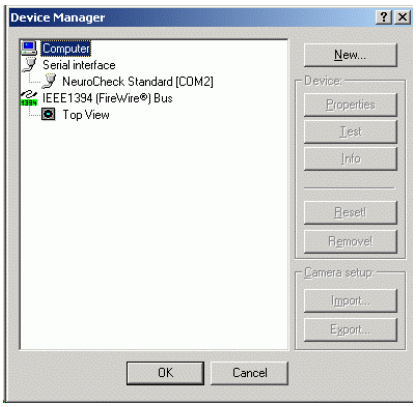
Now NeuroCheck is ready to exchange data via serial interface.

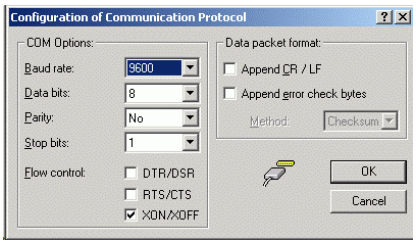




❶

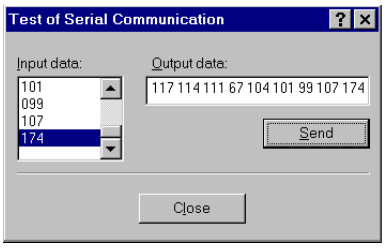
From this dialog box you can load the device driver for serial communication.

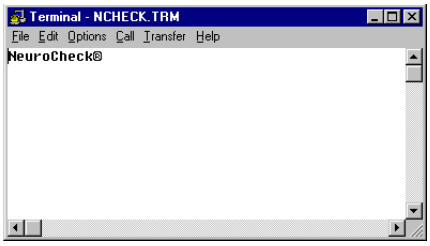




❷

Typical settings for serial communication with a terminal program.





❸

Test string as sequence of decimal ASCII codes in the test dialog and as plain text in the terminal program window.

9.5 Data Output using the Serial Interface

Problem

You want NeuroCheck to transfer results of check functions via the serial interface.

Result

You know how to configure output of results from specific check functions and send them via the serial interface.

Solution

❶ Load the check routine configured in chapter 2.5 for reading a bar code from a camera image. From the **Check Routine** menu choose **Data Output ▸ Serial Interface...**.

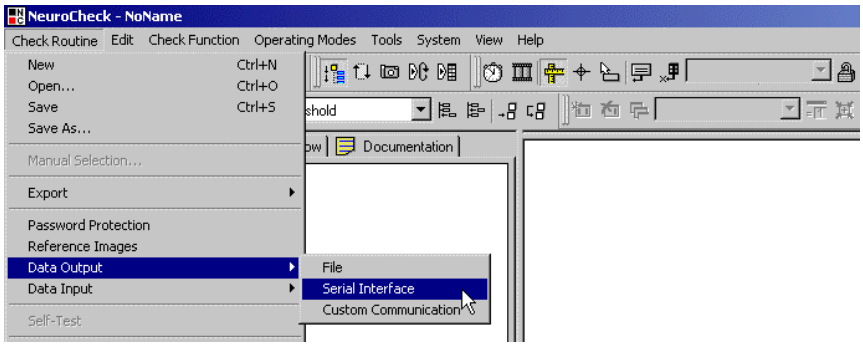
❷ In the **Serial Output Settings** dialog activate the **Generate serial output** check box to activate data output via serial interface globally. Then choose **Options**. On page **Administrative information** of the **Options for Serial Communication** dialog activate all check boxes.
Note: The bottom check box, **Include final result**, cannot be deactivated, because this would conflict with the setting on the **Output signals** page of the **Remote Control** dialog.
Confirm both dialogs with **OK**.

❸ Switch to the **Output** page of the edit pane. Destination **Serial interface** has already been activated globally. Now you have to activate it for function Identify bar code by clicking the check box in front of **Serial interface** below this function.

❹ In automatic mode NeuroCheck now waits for a start signal via serial interface according to the settings in the **Remote Control** dialog box. In hexadecimal notation the start signal is “0x02 0x53 0x03”. If your terminal program does not allow entering the control characters framing the S (ASCII “0x53”), you can use the `nc_start.txt` file included with this training course.

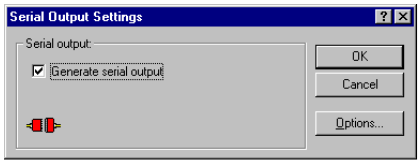
When NeuroCheck receives this start signal it executes one inspection run. Check function Identify bar code sends the bar code via the serial interface. According to the setting in the **Options for Serial Output** dialog box, the index of check and check function (counted from 0) as well as the check function identification number are appended to the check function result string. Function Identify bar code is the fourth function in the first check, hence the check index is 0, the function index 3. The identification number of function Identify bar code is 536 (see check function, identification in the index of the online help system).

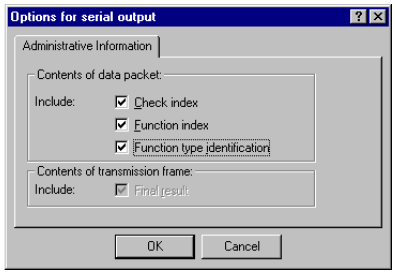
Furthermore the final check result is appended to each transmission. The check result ‘O.K.’ is represented by the string “ppp” for “passed”, ‘not O.K.’ by “fff” for “failed”.



❶

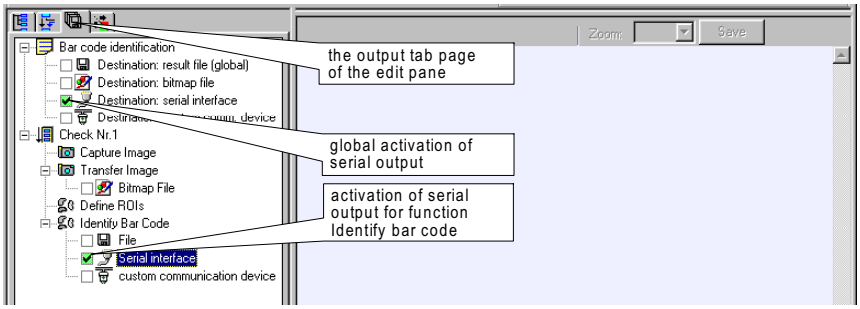
From the **Check Routine** menu choose **Data Output** ▸ **Serial Interface...**.





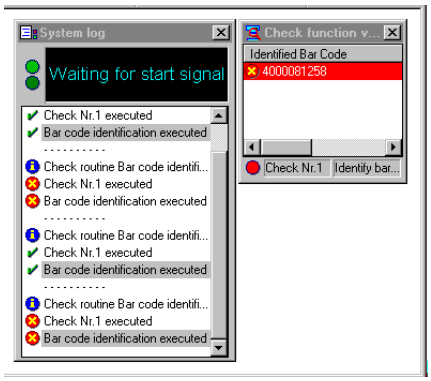
❷

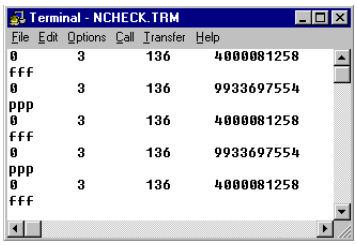
Activate output to serial interface and all administrative information items.



❸ 

On the output tab page of the edit pane activate serial output of check function results Identify bar code.



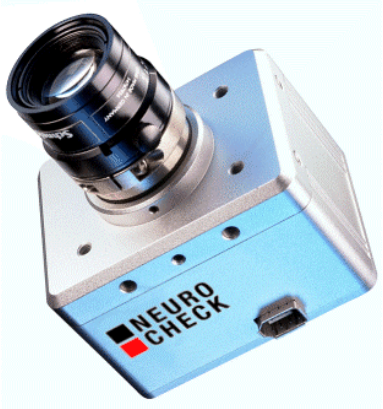


❹ 

NeuroCheck represents the check result ‘O.K.’ by “ppp” for “passed”, the result ‘not O.K.’ correspondingly by “fff” for “failed”.

10 Appendix: Images

The following pages contain the images used in the previous sections to enable you to work from a camera without having to worry about availability and illumination of real test pieces. They are also included as bitmap files in every complete installation of NeuroCheck, in the `Training` directory of the default path of NeuroCheck.



FireWire camera

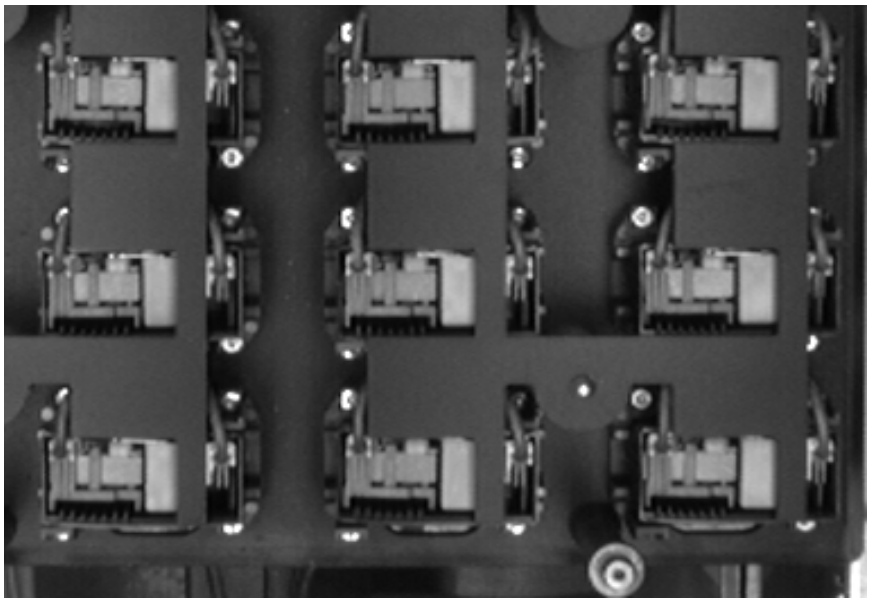
10.1 Bar Codes

Different bar codes as used in section 2.5, **Camera Images in a Check Routine**.

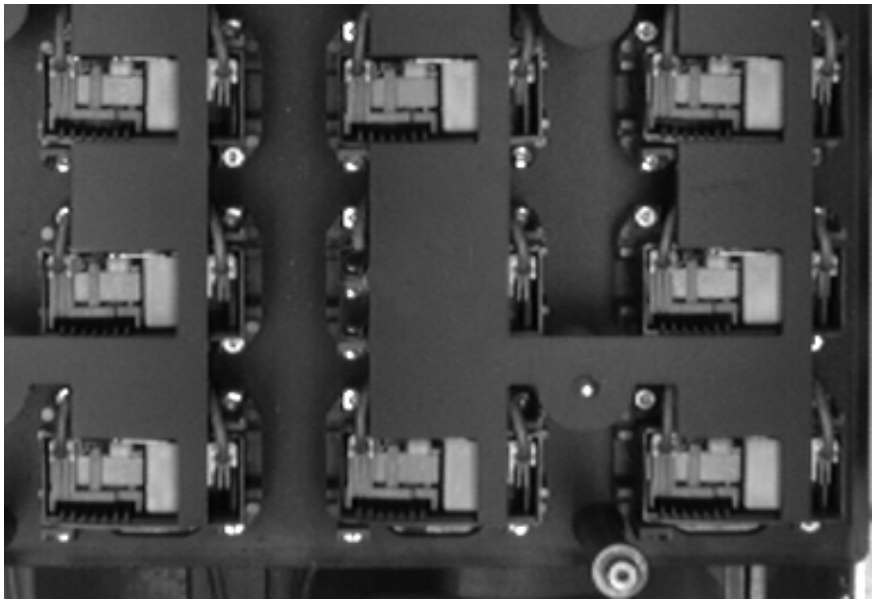
10.2 Demonstration Bitmap

Demonstration bitmap

10.3 Circuit board for template matching



Circuit board with nine elements



Circuit board with one element missing

10.4 Disks

3.5" disk, not write-protected

3.5" disk, write-protected

Disk with white spots

Disk with open slider

Disk with dislocated drive plate

Disk with dislocated slider

Completely destroyed disk