Image Stack Acquisition

SE-AS-075-EN

4. December 2017

Author: NeuroCheck GmbH, E-mail: support@neurocheck.com

Contents: This white paper describes the capture of so-called image stacks using NeuroCheck digital cameras with NeuroCheck 6.0/6.1 software.
Image stack acquisition is a special image capturing mode enabling the acquisition of several images within a short period of time and evaluating the images acquired so far at the same time.

Note: This document is not part of the official product documentation of the NeuroCheck software.

The check routine examples are available in a separate ZIP file (this will be described in the last chapter of this white paper).

Contents:

1. Introduction

NeuroCheck 6.0/6.1 software has a special image capturing mode for advanced use cases: the so-called image stack acquisition.

This mode allows the capture of several images within a short period of time and evaluating the images acquired so far at the same time.

Image capture and evaluation operate completely independent of one another. This way, images can be captured very quickly and then be evaluated without delay.

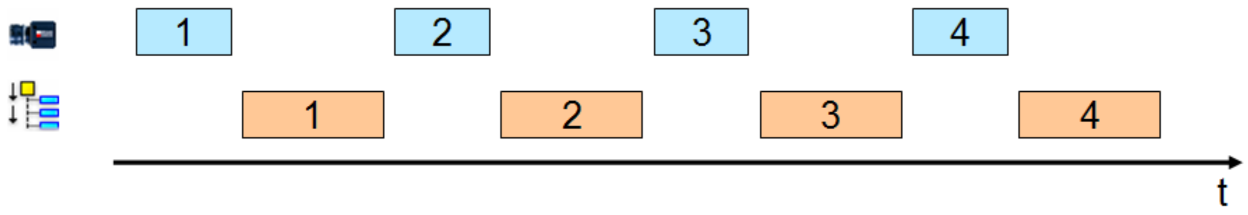
This special way of image capture is provided by the NeuroCheck 6.0/6.1 hardware driver for NeuroCheck digital camera series.

2. Motivation

This chapter describes first the image capturing methods available so far, and will then explain the need for the additional image capturing mode.

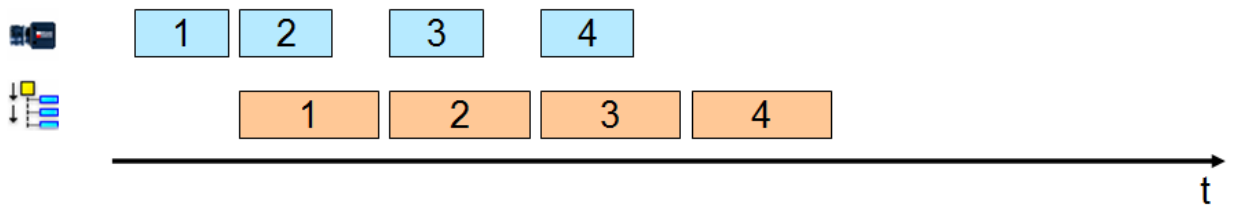
(1) Capturing and evaluating images in turn

The standard in NeuroCheck is to capture an image, evaluate it, and then capture the next image.



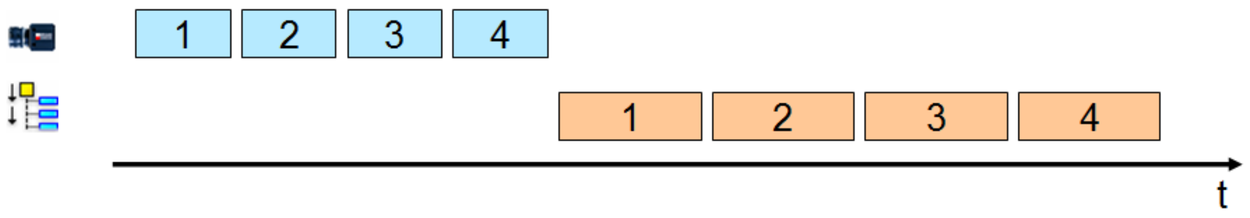
(2) Capturing the next image while evaluating the previous one simultaneously

Some applications cannot be implemented using the first method since the time available is not sufficient for image capture and evaluation. Therefore, NeuroCheck allows you to capture images in parallel mode, i.e. while one image is being captured, the previous image is evaluated.



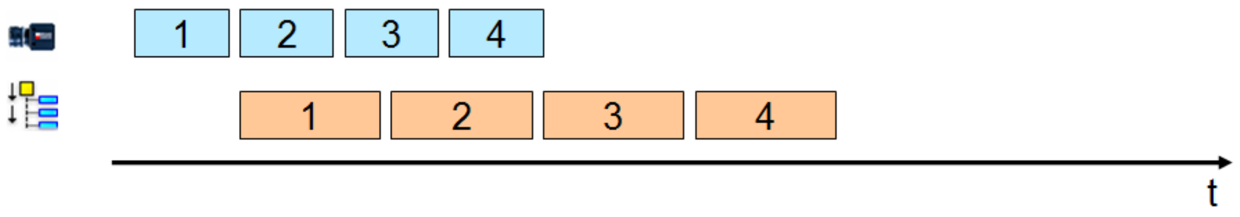
(3) Capturing a rapid image sequence and evaluating them subsequently

In some special cases it may be necessary to capture several images in rapid sequence so that there is no time for evaluation in between. This is the case, for example, when the objects are only a short time in front of the camera. In such a case the images are first transferred to the image tray and, after all images have been captured, are then loaded from the image tray and evaluated.



(4) New: Image stack acquisition: Capturing an image sequence while evaluating the captured images

For applications where there is only a short timespan for capturing the images and the time between the last image of a sequence and the first image of the next sequence is not long enough to evaluate all captured images, NeuroCheck has introduced a new way to capture images: image stack acquisition.



3. Image stack acquisition concept

The basic idea of image stack acquisition is to capture a fixed number of images with one camera while at the same time the check routine is executed independently.

For this, the image capturing is executed in a background thread. While the check routine is executed, the captured images of the stack can already be accessed even if not all images of the stack have been captured yet.

It is even possible to apply image stack acquisition to several NeuroCheck cameras on a vision system at the same time, independently of each other.

Image stack acquisition is a special function of the NeuroCheck 6.0 and 6.1 camera driver for NeuroCheck NCG and NCF digital cameras, and is thus not available for other camera types.

Please don’t confound the image stack acquisition concept of the camera driver with the image tray provided by the NeuroCheck standard software that enables you to store several images as well.

Please be aware that using an image stack with a large number of images will require a huge amount of computer memory.

4. Basic functionality

Control of the image stack capture takes place using the NeuroCheck standard check function *Control Device*.

To capture an image stack, at least three steps are necessary:

1. Determine number of images
2. Start capturing of image stack
3. Transfer captured images from stack

The necessary calls of *Control Device* are presented in this chapter.

4.1. Determine number of images

First you must set the number of images in the stack. To set the number of images in the stack, add the *Control Device* check function from the *Tools* category and open the parameter dialog. Select the camera you want to use for image stack acquisition. Select "Size of Image Stack" in the field *Set property* and enter the number of images you want to capture.

Please note:

- The size of the image stack must be set to a fixed value to enable image stack acquisition. Therefore, this check function has to be executed at least once before starting image stack acquisition.
- Executing the check function fails if an image stack acquisition has been started before and is as yet incomplete.
- The number of images can vary between 2 and 100.
- If you enter 0, the images captured for the image stack are erased.

4.2. Start capturing of image stack

After defining the image stack size, capturing images for the stack can begin. Capturing the image stack is started using the *Control Device* check function. For this, please add the check function, select a camera and select "Acquire Image Stack" in the field *Execute method* from the parameter dialog.

Each time this check function is executed, a new image stack acquisition is started and executed in the background.

Please note:

- The execution of the check function will fail if capture of the last image stack isn't completed by that time.
- Capturing the image stack is aborted if capture of one image exceeds the timeout value defined in the properties dialog of the camera. If your application has to wait a long time for a trigger signal, please deactivate timeout in the properties dialog of the camera.
- Executing the check functions *Capture Image, Capture Image in parallel* and *Control Image Acquisition* and opening the *Live Image Dialog* and the *Device Manager* abort capturing an image stack.
- Image stack acquisition is executed completely in the background, which is why there is no feedback in the user interface as to when the capture of the image stack will be completed. However, an entry is made into the NeuroCheck log file as soon as the image stack acquisition is completed (see chapter “Means of diagnosis”).

4.3. Transfer image data to NeuroCheck's run-time data pool

The third and last step is to select the image data and have them transferred into the NeuroCheck run-time data pool by the driver. While the two previous steps are usually executed just once per check routine run to start the image stack capture, the third step is executed for each image thus several times. To transfer the image, the image to be transferred has to be selected from the image stack.

Selecting the image from the image stack is done using the zero-based index of the image. That means the first image on the stack has the index 0, the second has the index 1, the third has the index 2, and so on. You can set the index of the image using the check function *Control Device* by selecting "Active Index in Image Stack" and set the index.

Please note:

- If necessary, the check function will wait until the image with the right index has been captured. If the capturing of the image is delayed, e.g. because the trigger signal is missing, the check routine is not continued. This may look as if NeuroCheck is stuck or doesn't react.
- The execution of this check function fails if no image stack capture has been executed.

After this, the NeuroCheck 6.0/6.1 driver for NeuroCheck digital cameras supplies the data for the selected image which then can be transferred into the NeuroCheck run-time data pool using the check function *Transfer Image* in source mode *Camera*.

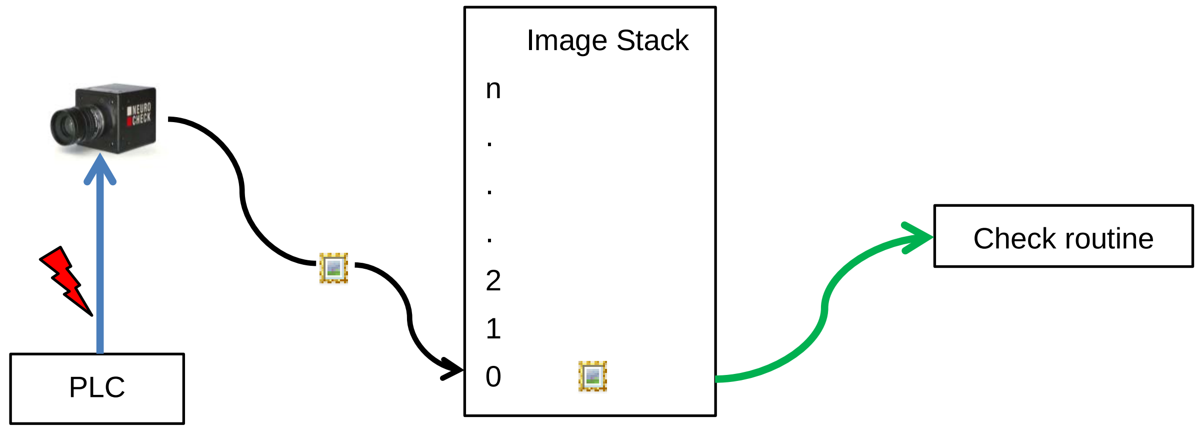
5. Advanced image stack options

5.1. Two timeouts

The image stack acquisition is a mechanism that is executed asynchronous to the Check Routine execution. The image stack is a transfer data structure between both processes: The image stack acquisition fills the image stack and the Check Routine accesses the image stack.

The asynchronous process of the image stack acquisition is started out of the Check Routine with the help of *Acquire Image Stack*. The camera is thereby waiting for a trigger-signal (marked blue in the following figure) and acquires an image if it notifies a trigger-signal. This is repeated until the amount of images that is defined by the parameter of *Size of Image Stack* in the Check Routine is reached. The camera waits a defined period of time for a trigger-signal. This timespan can be configured in the property dialog of the camera in the device manager of NeuroCheck.

It is possible to access a specific index in the image stack with the help of *Active Index in Image Stack*, visualized with the green arrow in the figure. However, it is not ensured that there is an image available at a specific index in any case. It is therefore recommended to define a timespan for which the Check Routine waits when no image is available. This can be done with *Timeout for setting active Index in Image Stack*.



Acquisition and supply of the image for the check depend on two timeouts that induce each other: If the acquisition fails, the image stack cannot be filled and consequently an access to an image at a specific index is not possible. The following table shows, which timeout hits in which case:

Case	**Which timeout hits?**
Acquisition fails.	The timeout with the smaller value hits.
Access to an index of the image stack that does not contain an image.	Timeout “Control Device“

5.2. Configure timeout for setting image index

In the simple check routine described so far, the check function *Control Device* waits at "Active Image in Image Stack" until the image has been captured. Sometimes it may be advantageous to terminate waiting after a timeout. For this you have to add the check function *Control Device* and set the property "Timeout for setting active Index in Image Stack". The set value is the time in milliseconds after which waiting for the image capture is aborted.

Please note:

- Executing the check function fails if an image stack is being captured.
- The set timeout applies to every call of "Active Index in Image Stack" via *Control Device*.
- Valid timeout values are between 1 ms and 99999 ms. For an infinite timeout, please set 0 as the value.

The timeout for setting the image index should be configured during initialization of image stack acquisition.

5.3. Abort image stack acquisition

For various reasons it may happen that less trigger signals are sent during a series than have been configured for the image stack size. In such a case the image stack acquisition will wait in vain for the last trigger signal and thus for the last image. As long as an image stack acquisition isn't completed, a new image stack acquisition cannot be started. In such a case it is necessary to abort the incomplete image stack acquisition. For this the check function *Control Device* must call the operation "Abort Image Stack Acquisition".

Please note:

- Executing the check function will fail if no image stack is or was being acquired.

If the image stack acquisition has already been completed when the operation "Abort Image Stack Acquisition" is called, the operation won't do anything. It is therefore recommended to call the operation "Abort Image Stack Acquisition" every time at the end of a check routine to make sure that a new image stack acquisition can be started with the next execution of the check routine. Since the final action is always executed independently of the check routine run, it makes sense to abort image stack acquisition with the final action.

6. Means of diagnosis

Since image stack acquisition takes place in the background, there is no visual feedback about the status of the image stack acquisition, neither in manual nor in automatic mode. To be able to follow the run of an image stack acquisition nonetheless, information is written to the NeuroCheck log file. Since image stack acquisition is a function of the hardware driver for NeuroCheck it is necessary to activate both the output of log message in the About dialog of the driver and the creation of a log file in the software settings of NeuroCheck.

Description of log file entries:

SetDevicePropertyValue: Set SizeOfImageStack to 10.

> The size of the image stack has been set, in this case to 10.

SetDevicePropertyValue: Set IndexTimeOut to 200.

> The timeout for setting the image index was set, in this case to 200ms.

Start acquiring a series of 10 images using background thread...

> A new image stack acquisition has been started, in this example with a scope of 10 images.

10 images acquired successfully in 475 ms.

> The image stack was captured completely. In this case the image stack consists of 10 images captured in 475 ms.

SetDevicePropertyValue: Set ActiveIndexInImageStack to 0.

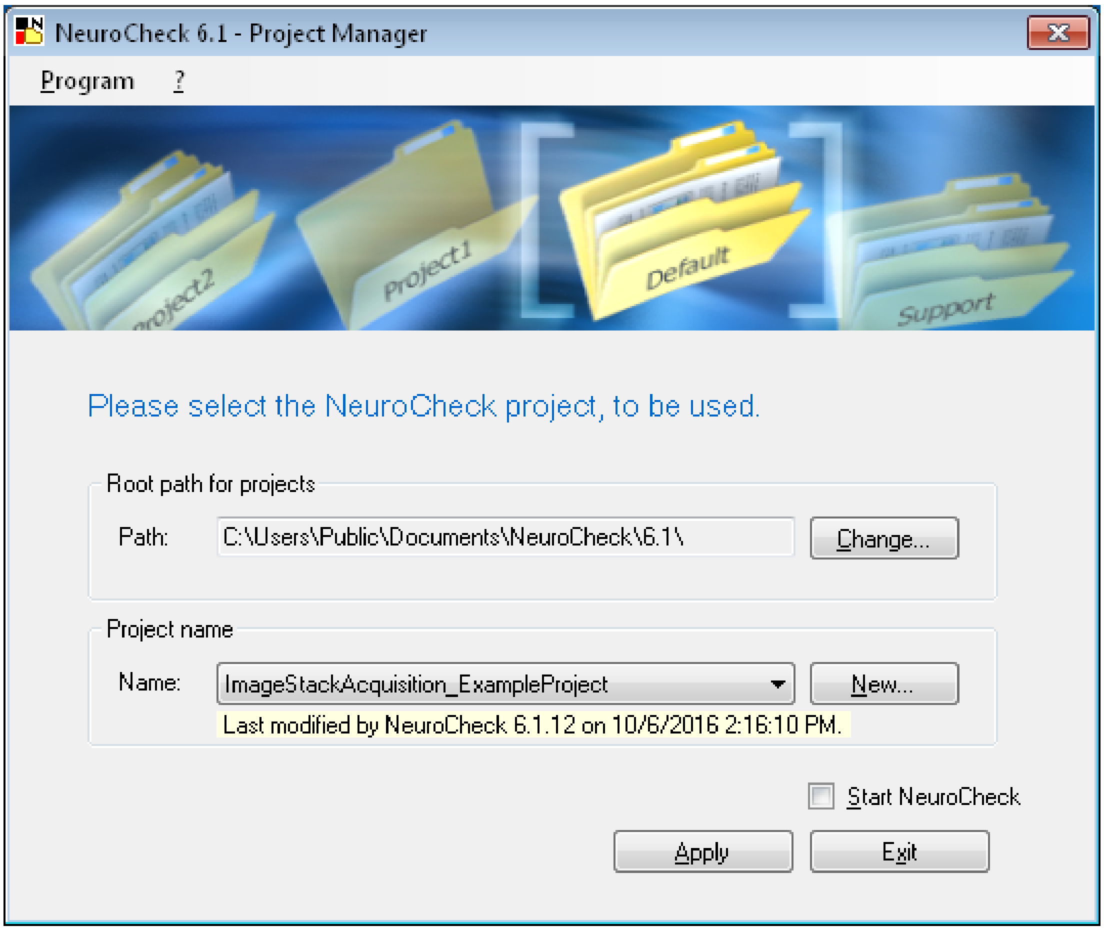
> An image was selected from the image stack. This image is now transferred to the NeuroCheck run-time data pool using "Transfer image". In this case it's the first image with the index 0.

Image stack execution aborted upon user request.

> An image stack acquisition was aborted.

7. Check routine examples

In a separate ZIP file with the name `ImageStackAcquisition_ExampleProject.zip` are several check routine examples available. We suggest that you unpack this file, copy its content into your NeuroCheck 6.1 project folder as a new project, and assign this project in the NeuroCheck 6.1 project manager:



Please select the NeuroCheck project, to be used.

Root path for projects

Path: C:\Users\Public\Documents\NeuroCheck\6.1\

Project name

Name: ImageStackAcquisition_ExampleProject

Last modified by NeuroCheck 6.1.12 on 10/6/2016 2:16:10 PM.

When starting NeuroCheck 6.1, you should immediately open NeuroCheck device manager and replace the camera type configured in the sample project by your actual camera.

After that you can load the example check routines. All sample check routines use the image stack acquisition, but differ in the way they process the acquired images.

7.1. Simple check routine

Sample check routine *„01-Simple Check Routine.chr“* shows the most simplest way to design an image stack acquisition with NeuroCheck. Each image is processed in a separate individual check. The necessary steps will be explained in the following, you can find these as individual checks in the sample check routine:

1. Initialization of image stack acquisition takes place in the start action. During initializing, the number of images is determined first ("Size of Image Stack"), then capture of the image stack is started ("Acquire Image Stack").
2. to 6.
   During the following individual checks, the captured images are evaluated individually. Each check starts by setting the index of the image about to be processed ("Active Index in Image Stack"). This check function will wait until the image with the correct index has been captured. Using *Transfer Image*, the image will then be supplied to the run-time data pool to be evaluated.

7.2. Check routines to process the acquired images with loops

Usually you will want to process all acquired images in the same way. Thus it is a good idea to do the processing in a single individual check and process it multiple in a loop.

7.2.1. Check routine with loop

The second sample check routine *„02-Check Routine with Loop.chr“* shows the easiest way to design a check routine with image processing in a loop. It consists of three individual checks, where the last check „Transfer and evaluation of images“ will be executed several times.

To be able to implement the loop, for this check routine two data registers of type “Integer value” had to be created in the “Input register” category. The first data register has the purpose to set the index in image stack dynamically during loop execution. The second data register defines the maximum number of loop executions, it corresponds to the total number of images to be processed.

The three individual checks in this check routine sample are designed this way:

1. *Initialization of Image Stack Acquisition.* The number of images to be acquired is set from data register „Number of Images“ by using dynamic data input.
2. *Initialization of data register*. In this individual check the data register “Loop Index” is initialized with the value 0.
3. *Transfer and evaluation of images*. This individual check will be executed multiple because on its property page it has been configured to run in a loop. The number of loop executions corresponds to the „Size of Image Stack“, it is controlled by using dynamic data input. Inside this individual check the data register value representing the loop counter is incremented by 1 by using check function „Modify Data in Register“. The data input of the check function that tells the camera device the index of the next image to be transferred is configured in a way that this value will be taken from the data register that represents the loop counter.

7.2.2. Check routine with loop – Exit Strategy

The third sample check routine *„03-Check Routine with Loop – Exit Strategy.chr“* provides the possibility to exit from the loop regulated by an input signal. To achieve this, you need an additional data register of type “Boolean”. This data register has been connected to a signal from a PLC. At the beginning of each loop execution, this signal is read. Only if this signal is present, check function „Read Data into Register -- Check if signal is present“ will yield “OK” and thus only then the image will be provided for evaluation by check function „Transfer image”. If the check function yields “not OK” the loop will break at this point.

This exit strategy can be used to send an exit signal from the PLC or master computer to force a premature exit from image stack acquisition. For instance, this can be used in case of a camera failure when no images can be acquired any more.

7.3. Check routine with GOTO

In NeuroCheck there is another possibility to design multiple execution of individual checks within a check routine: The GOTO instruction. Using GOTO, the flow behavior of individual checks is changed in a way that they will not be executed one after the other, but it is possible to jump after execution of an individual check to an arbitrary individual check inside the check routine, even to jump back to an individual check executed before. This way you can design a “loop” over several individual checks. To pass the current image from one individual check to another, the NeuroCheck standard image tray is used in the following examples.

7.3.1. Check routine with GOTO

The fourth sample check routine *„04-Check Routine with GOTO.chr“* shows a check routine design making use of the GOTO instruction. It contains the following individual checks:

1. *Initialization of Image Stack Acquisition.*
2. *Initialization of data register*.
3. *Transfer of current image*. Access image stack at current loop index and transfer of this image to the NeuroCheck image tray.
4. *Evaluation of image.* Copies the image from the image tray and evaluates it.
5. *Loop condition.* Increments the data register and checks if its value already corresponds to the total number of images. The flow property of this individual check has been adapted in the property dialog so that after an “OK” execution the individual check’s flow execution jumps back to the third individual check „*Transfer image*“.

This check routine sample uses a data register as loop index. This data register is used by dynamic data input for setting the current image index, for copying the image into the image tray and for copying it back from the image tray.

By using the GOTO instruction it is possible to separate provision and evaluation of the images into separate individual checks. This enables for a better structuring of the check routine on individual check level.

7.3.2. Check routine with – Exit Strategy

The fifth sample check routine *„05-Check Routine with GOTO – Exit Strategy.chr“* shows a check routine design providing the possibility to exit from the loop during execution. Similar to the exit strategy for loops, in this example an additional data register of type “Boolean” has been added and connected to a signal from the PLC. In individual check „Check for loop exit“ this signal is read and evaluated. The flow property of this individual check has been adapted in the property dialog so that the whole check routine execution is quit if the signal is not present.

Again, this example shows that using the GOTO instruction allows for a better structuring of different tasks into different individual checks.

7.4. Check routine with options

The sixth sample check routine *„06-Simple Check Routine with Options.chr“* enhances the first sample check routine by integrating the options shown in chapter “Advanced image stack options”. The configuration of the timeout for setting the image index was complemented in the start action. Aborting image stack acquisition was introduced in the final action to make sure the image stack acquisition is aborted if it hasn't been completed at the end of the check routine execution.