VLAIO-TETRA
Machine Vision for Quality Control
Demonstrators

Contents

1 Introduction

One of the deliverables of the Machine Vision for Quality Control project is to develop two demonstrators that can be used to educate students and companies in using the proper machine vision techniques for several applications. This document describes the design requirements and results for these two machine vision demonstrators. The core aims of the demonstrators are:

- Usage in practical workshops as a training setup
- Usage in educational activities
- Usage for fast proofs of concepts

These demonstrators aim to show a broad variety of machine vision applications and solution techniques using both commercial and open source hard- and software. There is also taken care of a proper balance between classical non data-driven machine vision and data-driven machine vision.

2 Demonstrator 1

This demonstrator showcases four machine vision applications and is shown in Figure 1. The demonstrator aims at showing a large variety of machine vision applications as it includes a 3D application, a measuring application, a color application, and a defect classification application. It also aims at using a variety on lighting principles such as a back light, a dome light, and a ring light. The demonstrator also shows hard- and software of a variety in machine vision vendors such as OMRON, Cognex, IDS, and Intel. The following sections elaborate on the four setups. All software for this demonstrator is to be found in https://github.com/MatthiasDR96/vision_demonstrator.git.

Figure 1: Demonstrator 1

2.1 Demo 1 - Detection of a ball in 3D

This demo shows an application of a ball that gets detected and of which the 3D coordinate in a specific coordinate frame gets determined. The ball is detected using classical image processing techniques such as thresholding and contour detection which outputs the pixel coordinate of the center of the ball in the image. Using camera calibration techniques, the pixel coordinate of the ball is transformed to metric coordinates in a coordinate frame defined by a chessboard calibration tool.

2.1.1 Hardware

The hardware that is used for this application only uses a RealSense D415 stereo vision camera with no additional lighting. This camera outputs an RGB image with a 1920 x 1080 (2MP) resolution at 30 frames per second and a depth image with a 1280 x 720 resolution at 90 frames per second. The camera has an USB3.0 interface. Also a blue ball is used as well as a checkerboard. The whole setup can be seen in Figure 2.

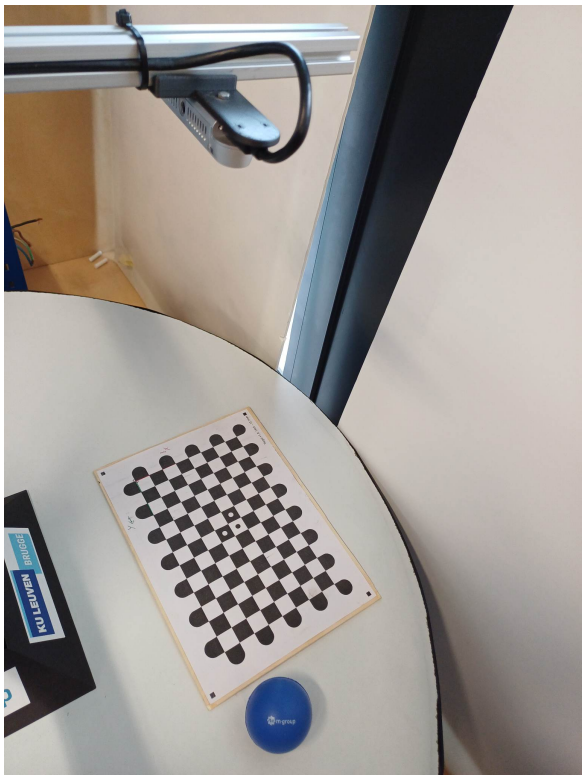


Figure 2: Demo 1

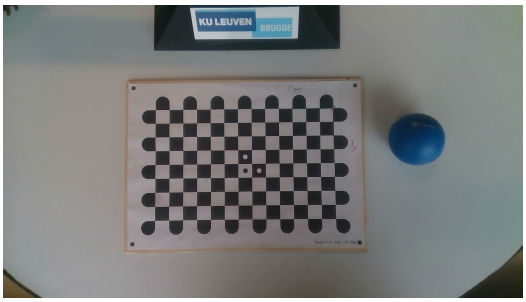
2.1.2 Software

The software that is used for this application is Python 3.9. The next sections will elaborate on the steps taken to obtain the result.

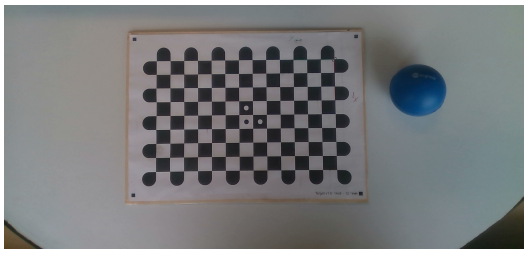
Image cropping A first step is to read the raw image from the camera. The raw image can be seen in Figure 3a. In the raw image, also the logo of KU Leuven is visible. As this has roughly the same color as the blue ball we want to detect, the image is cropped to remove the logo. In general, it is always a good idea to remove any background that does not belong to the item to be inspected. The cropped image is visible in Figure 3b.

Image smoothing A second step is to smooth the image using a Gaussian filter of 7 by 7. This to reduce noise in the image. The filtered image is visible in Figure 4.

Image color thresholding A second step is to segment the ball from the scene. This is done using thresholding where the color of the ball is used to theshold the ball from the scene. The image is first converted from RGB to HSV in order to facilitate the thesholding process. In RGB, all three color values show a high variance in different lighting conditions. This means that the ability of detecting the ball would change with changing lighting conditions, which is not appropriate. In HSV, the channel values are more invariant to changing lighting conditions which makes the detection of



(a) Demo1 - Raw image



(b) Demo1 - Cropped image

Figure 3: Demo 1 - Raw and cropped image

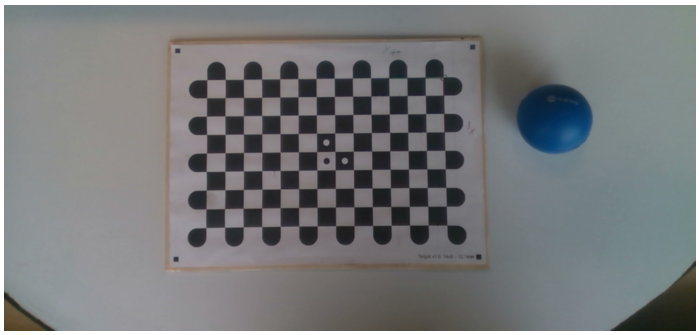


Figure 4: Demo1 - Filtered image

the ball more robust. The HSV values to threshold on are obtained via a visual calibration where the minimal and maximal HSV values are obtained such that the ball gets segmented from the scene. The resulting image after thesholding is visible in Figure 5.

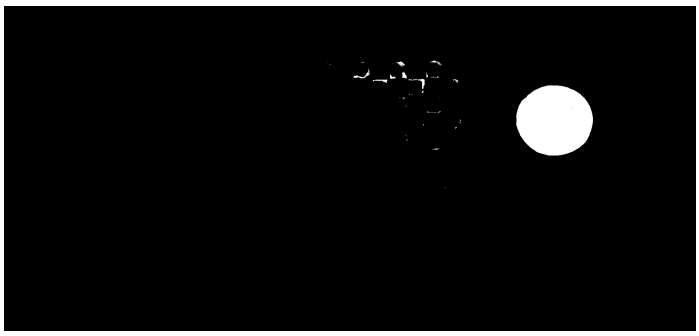


Figure 5: Demo 1 - Thresholded image

After thresholding the color, it can be possible that there are other features in the image of which the color also lies within the threshold range and that therefore also appear as white in the resulting image. Several techniques exist to remove these features and only keep the object of interest. The techniques that is used here is a morphological technique called closing, which is a

subsequent operation of eroding and dilating. The result after eroding is visible in Figure 6a. The result after dilating is visible in Figure 6b. These operation can be tuned to give a better result than the one we obtained.

(a) Demo 1 - Eroded image

(b) Demo 1 - Dilated image

Figure 6: Demo1 - Eroded and dilated image

Contour detection The previous step results in a binary image where the ball is shown in white pixels and almost all of the background in black pixels. A contour detection algorithm finds closed contours in such a binary image based on value changes from black to white. This algorithm outputs a list of pixels that make part of the border of the ball. Using this list, the center pixel coordinate of the contour can be found. To make sure that only the ball gets detected and nothing in the background, the area of the contours is calculated and only the items with an area above a certain value are included to be sure that the area is large enough to be the ball. The result can be seen in Figure 7. Beside the area of the contour, also the pixel coordinate of its center can be retrieved. This pixel coordinate will be transformed to a metric coordinate in a known coordinate system to get the known position of the ball in 3D.

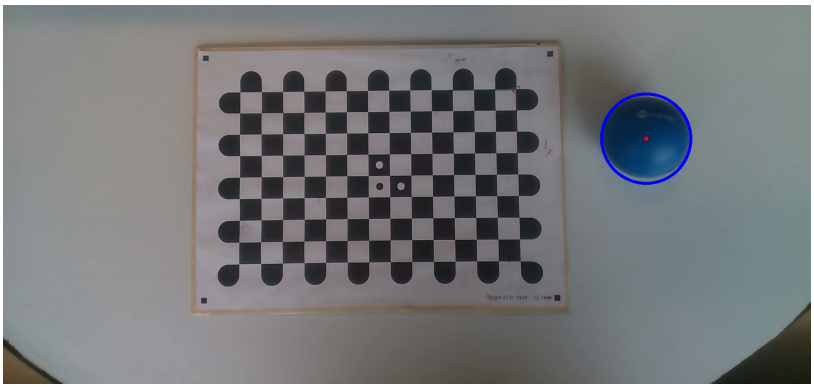


Figure 7: Demo 1 - Detected ball center

Intrinsic calibration The previous step resulted in the pixel coordinate of the center of the ball. Using an intrinsic camera matrix, the pixel coordinate gets transformed into metric coordinates with respect to the camera. This coordinate denotes where the center of the ball lies with respect to the camera. This intrinsic camera matrix contains the camera parameters such as focal length, pixel

depth, and camera center point. The intrinsic camera matrix can be obtained using a chessboard pattern of which the metric coordinates of the, in this case 126, black-black transitions are known (as we know the exact size of one square in meters). Using a computer vision algorithm, also the pixel coordinates of these black-black transitions can be obtained. Using both coordinate data, a system of linear equations can be solved to find the intrinsic camera matrix. The detection of the chessboard corners, together with the detected ball, is visible in Figure 8.

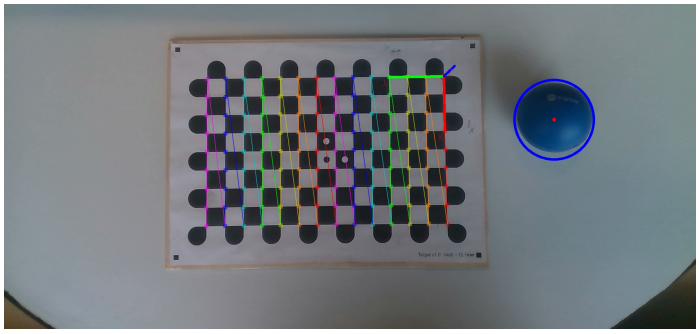


Figure 8: Demo 1 - Detected chessboard corners

Extrinsic calibration The previous step resulted in the metric 3D coordinate of the center of the ball with respect to the camera. Using an extrinsic camera matrix, the metric coordinate with respect to the camera gets transformed into a metric coordinate with respect to a user defined coordinate frame. This coordinate frame is defined using a chessboard calibration tool that defines the origin of the frame and the direction of its axes. This extrinsic camera matrix contains the transformation parameters that transform the camera frame to the chessboard frame and consists of a linear displacement x, y, z between these frames and a rotation matrix R between these frames. This final step results in the 3D coordinate in a user defined frame. Figure 9 shows the final result where the x,y, and z coordinates with respect to the user defined coordinate frame (which is represented by the chessboard) are printed on the image including the radius of the ball.

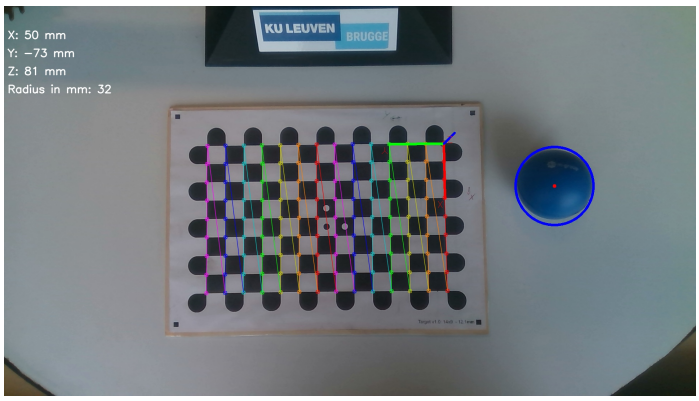


Figure 9: Demo 1 - Final result

2.2 Demo 2 - Measuring the saw angle of a sawblade

This demo shows an application of a sawblade of which the saw angle is obtained. The proper hardware components were selected to obtain an accurate measurement of the angle. The angle is measured using classical image processing techniques such as position compensation and edge detection. The whole setup can be seen in Figure 10.

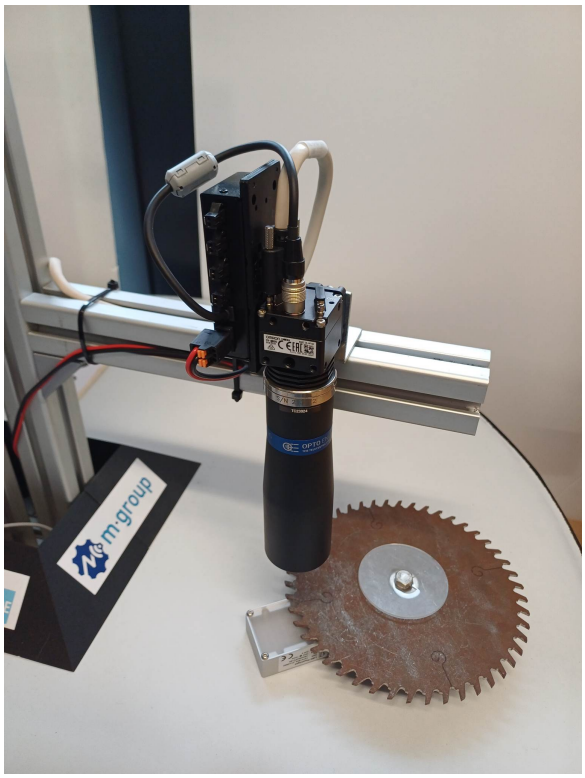


Figure 10: Demo 2

2.2.1 Hardware

The hardware that is used for this application includes:

- Camera sensor: An OMRON FH-SMX05 monochrome camera with a resolution of 2448 x 2048 (5MP) and an USB3.0 interface.
- Camera lens: A telecentric lens with a magnification of 0.35.
- Light: A red backlight.

The backlight is used to obtain an almost binary image of the sawblade with respect to the background. The telecentric lens is crucial in this measurement application as it prevents deformation of the image that would typically occur with a normal lens. This lens has an infinite focus which means that it keeps the light beams in parallel so that no image deformation occurs. A monochrome

camera is used to obtain a high accurate measurement. Color camera's could suffer from chromatic aberration and the color filtering and interpolation techniques in these camera's highly decrease the resolution of the image. To read the camera, control the light in strobing mode, and implementing the vision software, an FH5050 vision controller of OMRON is used. Also a FLVTCC4 light controller of OMRON is used to activate the light in strobing mode that is synced with the camera frame rate.

2.2.2 Software

The software that is used for this application is the vision software of OMRON. The software does not require any programming skills and allows the user to develop a machine vision application using some graphical tools and configuration of these different tools. The next sections will elaborate on the steps taken to obtain the result.

Image retrieval In a first step, the raw image is retrieved from the camera. Image retrieval is a function that can be used in the software and looks as in Figure 11.

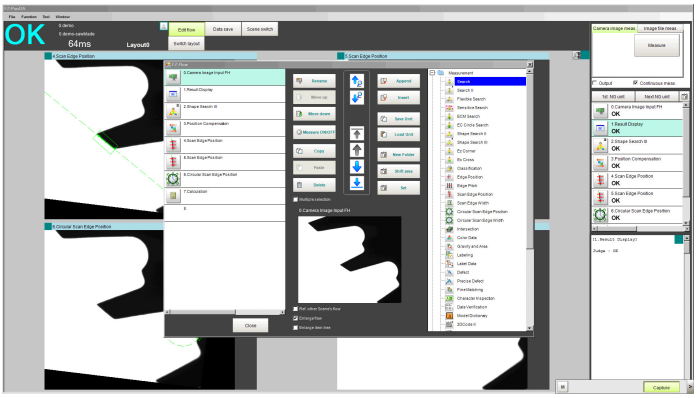


Figure 11: Demo 2 - Image retrieval

Image compensation The second step makes sure that the tooth of the sawblade is always in the same position in the image. To do so, the user draws a rectangle about the tooth so the algorithm can recognize the shape. Subsequently, at every new frame, the algorithm detects the shape and places it in the position of the initial defined tooth. In this way the tooth is always in the same position in the image and subsequent analysis can be done more easily. This function can be seen in Figure 12

Line detection On the tooth, two areas are defined a each side of the tooth where a line detection algorithm needs to detect a black to white transition. Using this transition, the algorithm can define a line. For both lines, the angle with respect to the horizontal is given as an output. From these angles, the angle of the sawblade can be calculated very easily. The tools to detect the edges are visible in Figure 13a and Figure 13b.

Circle detection Also the radius of the tooth base can be obtained. A tool to detect circles is available in the software and is depicted in Figure 14.

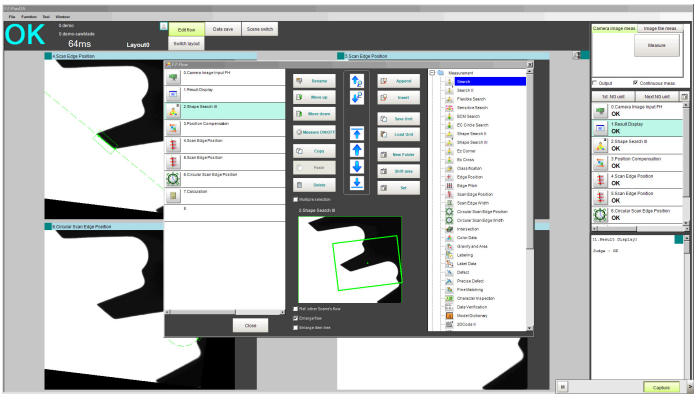
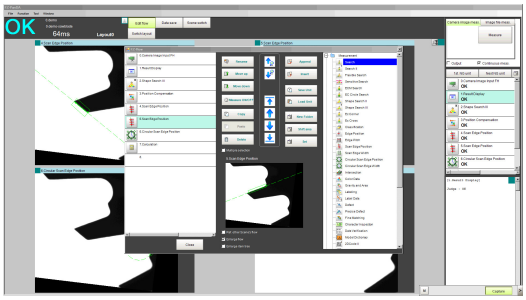
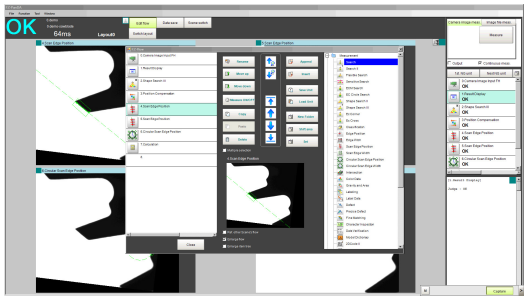


Figure 12: Demo 2 - Image compensation



(a) Demo 2 - Long edge detection



(b) Demo 2 - Short edge detection

Figure 13: Demo 2 - Edge detection

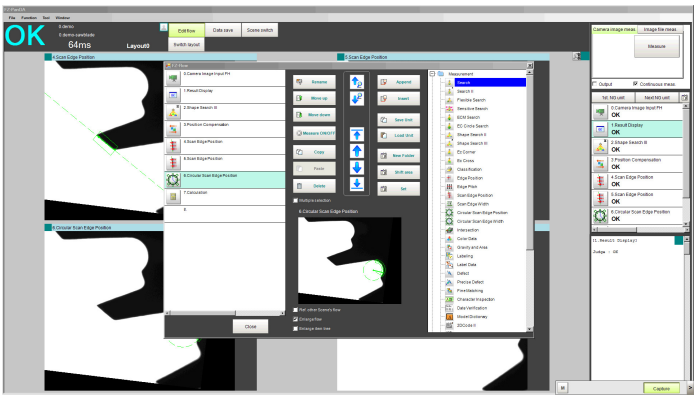


Figure 14: Demo 2 - Circle detection

Final result Finaly, all detections and calculated angle and radius can be shown on one image. This can be seen in Figure 15.

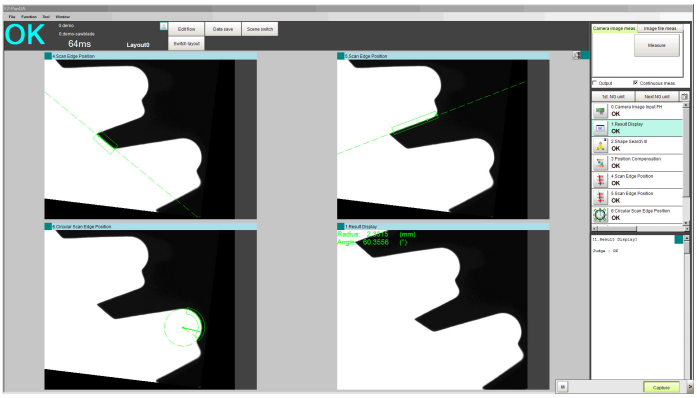


Figure 15: Demo 2 - Final result

2.3 Demo 3 - Determination of the value of a resistor

This demo shows an application of color detection of a resistor. The three core color bands of the resistor get detected as well as the color of these bands. Based on the color detection the value of the resistor can be determined. The bands are detected using classical image processing technique such as color thresholding and contour detection. The colors are obtained using machine learning to obtain a robust detection in varying circumstances. The whole setup can be seen in Figure 16.

2.3.1 Hardware

The hardware that is used for this application includes:

- Camera sensor: An IDS U3-3280CP-C-HQ Rev 2.2 color camera with a resolution of 2448 x 2048 (5MP) and an USB3.0 interface.
- Camera lens: A Ricoh FL-CC1614-5M 16mm lens with a resolution of 5MP.
- Light: An IDS5-00-250-1-W-24V white dome light.

The dome light is used to have a uniform light distribution on the resistor and to prevent glare on the resistor. It eliminates changing ambient light and provides a stable light condition that ensures a robust detection.

2.3.2 Software

The software that is used for this application is Python 3.9. The next sections will elaborate on the steps taken to obtain the result.

Resistor extraction At first, the raw image is extracted from the camera. This image is denoted in Figure 17a. I a first preprocessing stage, the resistor is segmented from the background. With the current light intensity and exposure time, the background is not fully saturated. A simple thresholding on a grayscale will remove all the saturated background pixels. The result after thresholding is shown in Figure 17b. After thresholding, some morphological transformations (opening and closing) are applied to the binary image resulting from the thresholding in order to remove the metal wires

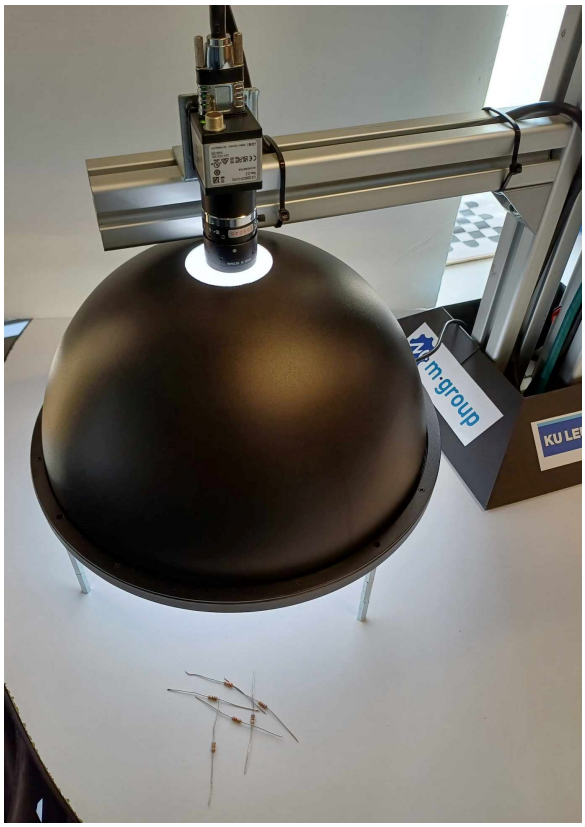
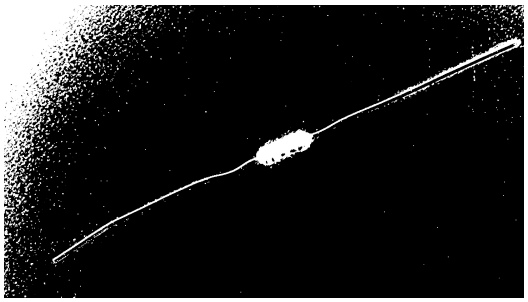


Figure 16: Demo 3

on each side of the resistor core. This results in a binary image where the resistor core is denoted in white pixels and the background in black pixels. This is visible in Figure 18a. A contour detection algorithm is used to detect the contour of the resistor. The contours are filtered on size so that the contours with an area larger than the resistor and the contours with an area smaller than the resistor are removes. What remains is the contour of the resistor. The contour is visible in Figure 18b. Using this contour, a bounding box is defined around the resistor and the longest axis of the box is obtained including the angle of this axis with respect to the horizontal. This angle is used to rotate the resistor so that it gets aligned with the horizontal. This makes sure that any resistor in any position with respect to the camera gets transformed to the same position for further analysis. The aligned resistor is shown in Figure 19.

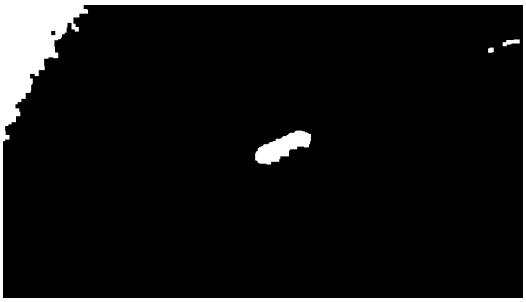
Bands extraction The previous steps resulted in a segmented resistor that is positioned horizontally. In this step, the color bands get extracted. This is done using simple thresholding on the HSV-representation of the image. This is possible because the dome light results in a stable lighting condition. The thresholding is done in two steps: (i) one mask is formed where the background gets removed and only the resistor body remains, (ii) the second mask is formed where the area in between the color bands gets removed and only the color bands and the background remain. Both are visible in Figure 20a and Figure 20b. Both masks are multiplied so that only the color bands remain. After some morphological transformations to be sure that only the three color bands remain and

(a) Demo 3 - Raw image

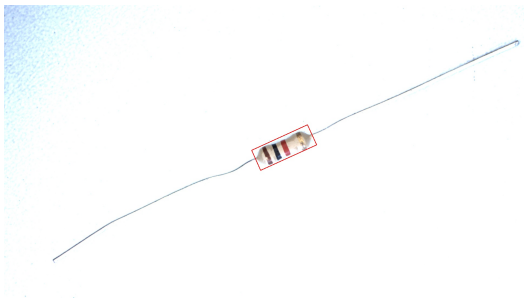


(b) Demo 3 - Background thresholding

Figure 17: Demo 3 - Background thresholding



(a) Demo 3 - Morphological transformation



(b) Demo 3 - Contour detection

Figure 18: Demo 3 - Contour detection

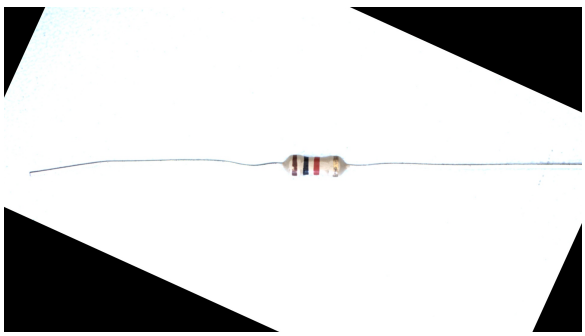


Figure 19: Demo 3 - Aligned resistor

noise is removed, a contour detection detects all three color bands. The thresholded colour bands and the result after morphological transformation are visible in respectively Figure 21a and Figure 21b. Based on the last result, the contours of the color bands are found by taking the first three contours counting from the left, ignoring the gold band. The contours are visible in Figure 22.

Color estimation For color estimation, a Machine Learning approach is used. Once the color bands are detected, the mean pixel value of all pixels within each contour is saved in a dataset

(a) Demo 3 - Background removal

(b) Demo 3 - Resistor color removal

Figure 20: Demo 3 - Color band extraction phase 1

(a) Demo 3 - Color bands before morphological transformation

(b) Demo 3 - Color bands after morphological transformation

Figure 21: Demo 3 - Color band extraction phase 2

Figure 22: Demo 3 - Contour detection

that contains three columns with the H-, S-, and V-values together with the ground truth label as a fourth column. During data generation, lots of images of different resistors are captured and labeled. This results in a large dataset containing labeled data of all resistors. A Support Vector Machine is trained on the dataset and learns what label to predict based on the HSV-values of the color bands. The dataset is visualized in Figure 23 where each point represents a color band represented as a HSV-coordinate. It is clear that the colors extracted from the color bands are easily separable and thus easy to learn by a Machine Learning algorithm. During the data validation, each of the above steps were repeated on each capture image of a new resistor and the predicted output was compared to the label. The algorithm reaches a 100% accuracy and is thus able to determine the resistor value of all resistors. The color codes are finally converted to a resistor value that is shown on the final image visible in Figure 24. The Figure shows a resistor to be labeled as 'zkr', where brown was labeled as 'z', black as 'k', and 'r' as red. Converted gives a value of 1k Ohms for the resistor in the image.

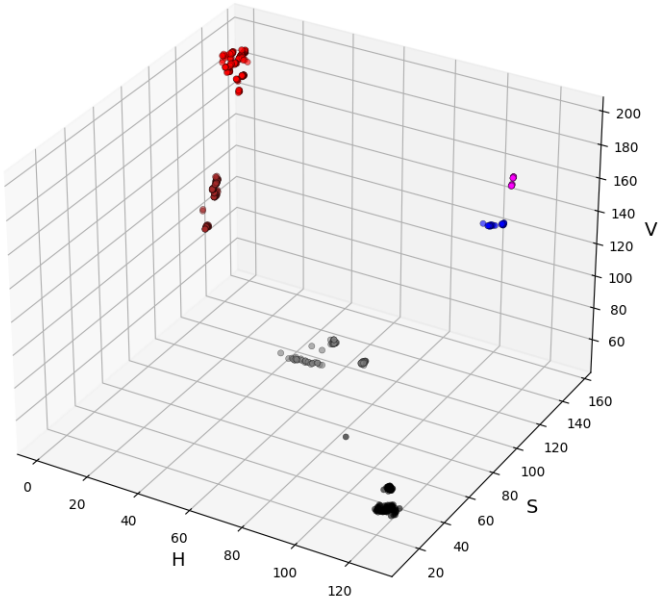


Figure 23: Demo 3 - Dataset

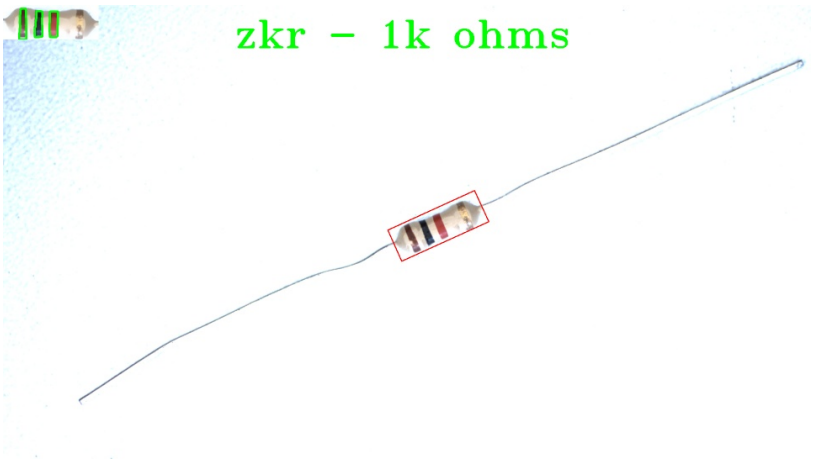


Figure 24: Demo 3 - Final result

2.4 Demo 4 - Classification of surface defects on wooden sheets

This demo shows an application of surface defects classification on wooden sheets. The samples show a sheet with a knot, a scratch, and with no defect. The defects are classified using data-driven machine vision techniques. More specific, a neural net was trained using transfer learning to obtain a classification of the defects. The whole setup can be seen in Figure 25.

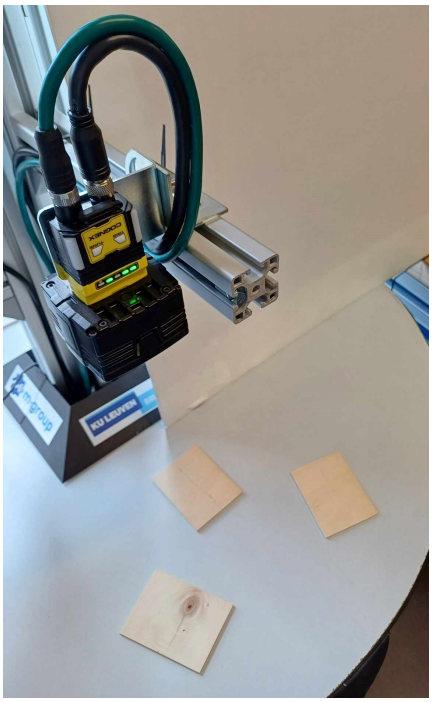


Figure 25: Demo 4

2.4.1 Hardware

The hardware that is used for this application is a Cognex in-sight 2800 smart camera with a multi-torch and a 12mm lens. This outputs an RGB-image with a resolution of 1440 x 1080 (1.6MP).

2.4.2 Software

The software that is used for this application is the In-Sight Vision Suite software of Cognex. The software does not require any programming skills and allows the user to develop a machine vision application using some graphical tools and configuration of these different tools. For this application we used the classification tool. Using this tool it is very easy to train a model to classify image in different classes. The classification tool is visible in Figure 26. The classes we have are 'Knot', 'Scratch', and 'No Defect'. The first to classes are referred to as bad objects, and the last class as a good object. Using the software it is possible to easily modify these classes and label new images. Images can be taken and the respective label can be assigned. The model is training each time a new image is added and also directly shows its prediction for every new picture. In total we made 36 pictures, 12 with a knot, 12 with a scratch, and 12 with no defect. The model manages to properly

classify the objects. Two objects that are classified as bad are shown in Figure 27a and Figure 27b. An object that is classified as good object is shown in Figure 28.

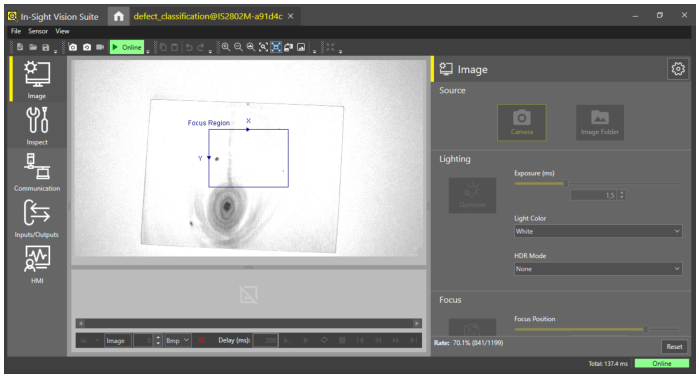
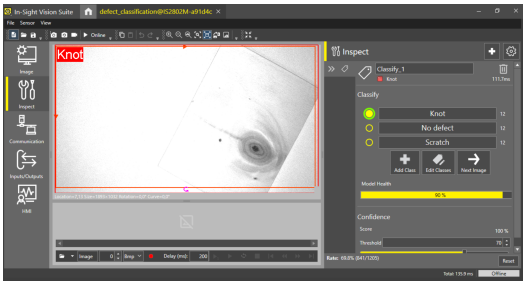
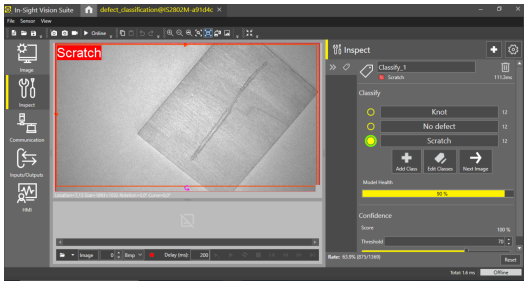


Figure 26: Demo 4 - Classification Tool



(a) Demo 4 - Knot



(b) Demo 4 - Scratch

Figure 27: Demo 4 - Bad objects

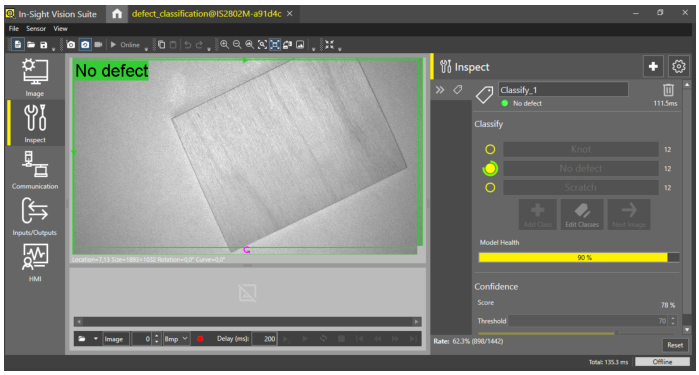


Figure 28: Demo 4 - No defect

3 Demonstrator 2

The second demonstrator focuses on industrial applications of machine vision algorithms and uses of the shelf components often used in production environments. This is a single demonstrator with several setups integrated to adapt as necessary. For example a different setup can be used during training (e.g. base setup) or an event where we would like to have a demonstration that is a bit more dynamic (e.g. high speed rotation). Several configurations are prepared and should be implemented in the final product:

- General purpose setup for training/ testing
- Rotation table for automatic demonstrations
- High speed setup for events
- Preparation for further expansions

3.1 Demo 1 - General purpose setup

The general purpose setup is intended to be the default configuration of the demonstrator. This configuration is the most versatile for both testing of lighting and camera types and generating Validation or training data for new projects. In this sections we describe the basis hardware for the setup, that will also be reused in the other demo's.

3.1.1 Hardware

Controller The central controller is an industrial PC based on a Quad core AMD processor running a version of windows 10 with a realtime kernel adaptation (Twincat 3). This controller has several interfaces that will be used in the project:

- Ethernet interface (programming, data server, etc)
- IO backplane (connecting hardware interfaces such as digital input or output and specialized functions like motor control)
- Standard computer interfaces (USB for data storage, DVI for screens, etc..)
- Gig-E interface for industrial camera's (2.5 gigabit possible)

Frame A frame has been designed as base for the other components and parts. By using aluminum extrusion profiles there is the option of adding several additional components as needed by using T slot bolt. Several key design point have been Incorporated:

- Modular base to mount the hardware, e.g. camera's, lighting, demonstration pieces, ...
  - Space for 2 camera's
  - Exchangeable base for different applications
  - Mounting points for lightting options
- Different components can be easily added or removed in the existing design

  - As lightweight and movable as possible
  - Fits in a standard car for transportation
- Provide insulation from external light sources

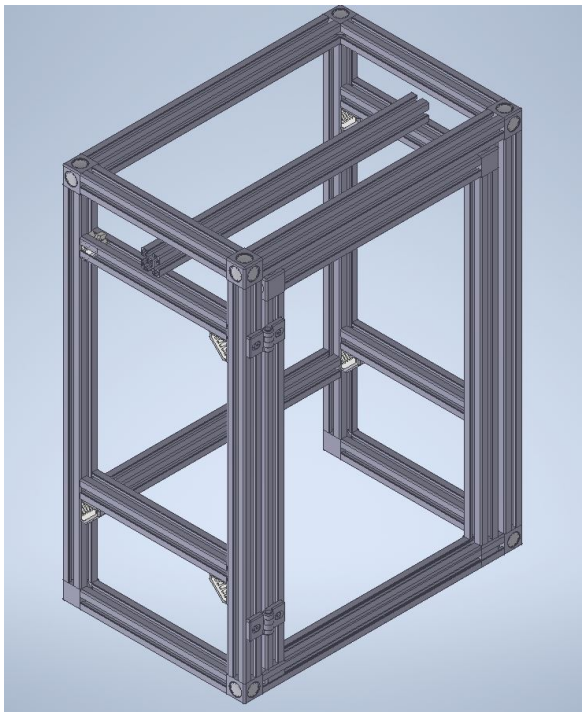


Figure 29: Early sketch of the basic frame for Demonstrator 2 without paneling

Camera and lens Camera's to be used for this setup are 2 basic ones. A black and white and a color version to be able to show the differences between them. These 2 camera will be used illustrate the differences in both data generation and light absorption.

The black and white camera is a Baumer VEXG-25M:

- 1920 x 1200 pixels
- 41 fps
- GigE-Vision compatible

The color camera is an IDS UI-5200SE-C-HQ:

- 4096 x 3000 pixels
- 8.6 fps
- GigE-Vision compatible

: 

Figure 30: Example of Side aligned lighting: courtesy of Vision doctor.com

This is not exactly a low end color camera, but will also be used in further demonstrators and imaging. If lower resolution images are needed we can downscale as needed. The main goal is to provide 2 totally different camera's, so we can test a potential new use. And visualize the differences between the technologies and application to select the best cost/benefit result.

Lighting solutions For the basis lighting we provide a ringlight, backlight and 2 barlights. This diverse setup is another preparation to demonstrate several different lighting solutions.

The ringlight is to illustrate the application that is relatively easy to setup (camera and light mounted on the same location, possibly from a single combined unit. While it provides a general and diffuse light, the source is always in the lightfield configuration. A multicolor version was selected to have a broad range of possible tests that can be run based on the samples used.

The bar lights can be used to implement arbitrary angles of lighting and can be used in both in lightfield or darkfield configurations.By mounting them on different locations we can demonstrate the impact of each of those positions. For example a scratch or ridge that is difficult to get an image of with good contrast can be clearly seen when the lighting is in a darkfield configuration. Adding the polarization filter to both the bar lights and the camera lens will also be added to show the possibilities in avoiding reflections.

When trying to get a clear silhouette of a component, for example for measurement applications, we can provide a backlight with a panel integrated in the bottom of the frame. This panel will also act as a base to deposit the test samples on (with a scratch resistant cover to prevent wear and tear).

3.1.2 Software

The focus of the demonstration software is both the ability to capture comparative pictures for validation purposes and provide a training set that can be used during applications training. Software is still to be written and tested. We will use the Twincat 3 environment as a base application for both machine control and XXX. This part will implement the following logical components:

- Image capture and storage
- Lighting control via IO/ ADS communication protocol

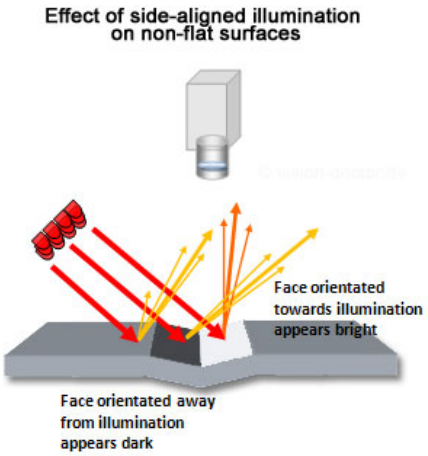


Figure 31: Illustration of the effect of side aligned lighting: courtesy of Vision doctor.com

3.2 Demo 2 - Rotation table

The rotation table is a setup intended for continuous demonstration. By adding ta rotating table we can run through several test pieces automatically while the person explaining the logic is free to focus on the person observing the demonstrator. The intended use is both longer and demonstrations for a broader audience and traings to a more advanced public.

3.2.1 Hardware

Design to order The rotating table and motor can be mounted in the frame from demonstrator 1 with relatively simple tools. The base is a motor linked to a rotating plate the is positioned between the 2 camera locations. So both camera's can be used at the same time, but on different samples.

The motor is a servo model with an absolute encoder that can be used the accurately position each of the samples in a repeatable position, avoiding the need for extra sensors on the spinning table. Speed is limited to ensure a safe operation of the setup.

Camera and lens The same camera's of the previous demonstrator will be used for this application as the intended image grabs are similar.

Lighting solutions Several lighting options are available, same as in the first demonstrator. In the first version the backlight will not be Incorporated to keep the design of the rotating table design simple. If needed a version can be conceived that includes a "window" of transparent material.

3.2.2 Software

Software to be written and tested. We will use the Twincat 3 environment as a base application for both machine controll and XXX. This part will implement the following logical components:

- Image capture triggerd by an extrernal impulse (encoder position)
- Lighting control based on the sample
- Visualization of the actives function

3.3 Demo 2bis - Camera system using transfer learning

An expansion on the second demonstrator is based around the integration of an intelligent camera. This camera would only do basic communication to the main controller and process the machine vision algorithms internally on the camera.

Hardware The hardware is still under discussion. We are looking into the 300O series from Cognex, but are limited by the budget of the project. As an alternative we have the 2800 series.

Software The software is linked to the camera using either a transfer learning system with a graphical interface. Or a spreadsheet based programming language.

3.4 Demo 3 - High speed demonstrator

3.4.1 Hardware

This configuration builds on the existing hardware, but add a bit of a show element by speeding up the process. IN this case we will focus on the geometries of a saw blade at a relatively high speed. The intended use is to draw attention and show a process at relevant speeds for industrial applications. Similar speed requirements might for example also be present in chip manufacturing or the food industry.

Designed to order hardware This demonstrator uses the hardware from the previous demo's 1 and 2 and adds a saw-blade and relevant safety devices.

The motors used in demonstrator 2 are already selected to allow a high speed rotation of the table. We will replace the table with a model that can safely mount a saw blade (or component strip) at reasonably high speeds. Practically this means a "catch" system that prevents the blade from leaving the setup in case of an issue and a protection that prevents the teeth from making contact with anything as a back-up.

Camera and lens The main focus in this case will be the high speed triggering of the camera. Therefore the design is a fully integrated one using everything within the same manufacturers ecosystem. While the data transfer will still be using the GiG-E interface, the triggering of both camera and lights will be done with a bus system utilizing a distributed clock system. The selected camera is the Beckhoff VCS2000-0200:

- 1930 x 1200 pixels
- 164 fps
- GigE-Vision compatible
- Fieldbuss based triggering

Note, this is a new product that may have some issues with availability at the launch. Therefore the the basis camera can stand in to a certain speed. If the camera would probe to be unavailable during the timeframe of the project we can explore other options (for example IDS has a similar offering, but integration will be more difficult).

Lighting solutions In this demo we will be using the backlight to create a silhouette of the sawblade. Due to the fast movement of the

3.4.2 Software

Software to be written and tested. We will use the Twincat 3 environment as a base application for both machine control and image evaluation. This part will implement the following logical components:

- Internal triggering of a high speed camera by the servo motor
- Synchronized lighting control via Ethercat protocol
- Measurement of features

3.5 Demo 4 - Interactive display test

This demonstrator mimics an application at the end of a manufacturing or refurbishing process where the product is checked against a list of pass/fail tests. In this case a simple LED display with some additional Leds is used to emulate this function. Commonly this is a time consuming and boring process for the operator. Which has a significant danger of leading to inaccuracies in the process. The controller applies a desired set-point and using the camera we check if the product does this in real life.

3.5.1 Hardware

Design to order The display has been supplied by the firm TVH as port of the project and is a very simple version controlled by a bit shift register.

Camera and lens The same color camera of the previous demonstrator will be used for this application as the intended image grabs are similar.

Lighting solutions Lighting will be minimal as the modules themselves produce light. Therefor we op the test with only internal illumination.

3.5.2 Software

Software to be written and tested. We will use the Twincat 3 environment as a base application for both machine controll and XXX. This part will implement the following logical components:

- Image capture triggered controller
- Interaction with the display
- Labeled storage of test pictures for future analysis

3.6 Demo 5 - Transfer learning on a Cognex system

A final demonstrator that will be available is the use of a Cognex camera model using transfer learning to do a classification function. Currently we are still in negotiation about the model that will be used. Once finished we will update this section.

4 Conclusion

Two demonstrators were developed in this project and both show a large variety on machine vision applications and techniques. Both non data-driven and data-driven techniques are showcased with the demonstrators and also a large variety on vendors are represented. The summary of the materials used for demonstrator one is shown in Table 2.

	Demo 1	Demo 2
Description	Detection of a ball in 3D	Measuring the saw angle of a sawblade
Application	3D application	Measuring application
Technique	Stereo vision	Line / circle detection
Camera	RealSense D415	Omron Monochrome
Light	/	Back light
Lens	Fixed focus	Telecentric

	Demo 3	Demo 4
Description	Determination of the value of a resistor	Classification of surface defects on wooden sheets
Application	Color application	Classification
Technique	Machine Learning	Deep Learning
Camera	IDS Color	Cognex in-sight 2800
Light	Dome light	Ring light
Lens	Fixed focus	Liquid lens

Table 1: Summary of components used in demonstrator 1

	Demo 1	Demo 2
Description	General purpose setup	Rotation table
Application	Data generation, measuring	Measuring application/ classification
Technique	Image capture	Triggered image capture, measurement
Camera	VEXG-25M / IDS UI-5200SE-C-HQ	VEXG-25M / IDS UI-5200SE-C-HQ
Light	As needed (Ring, bar, back, ...)	Ring or Bar light
Lens	Fixed focus	Fixed focus

	Demo 3	Demo 4
Description	High speed inline quality check of sawblade	Interactive checklist of LED display
Application	Blob detection	Circle detection, basic OCR
Technique	Blob detection	PLC control,
Camera	Beckhoff VCS2000-0200	IDS UI-5200SE-C-HQ
Light	Back light	None
Lens	Fixed focus/ telecentric test	Fixed focus

	Demo 5	
Description	Transfer learning demo	
Application	Automatic classification	
Technique	transfer learning	
Camera	Cognex series 2800/3800	
Light	Integrated light	
Lens	Fixed focus	

Table 2: Summary of components used in demonstrator 2