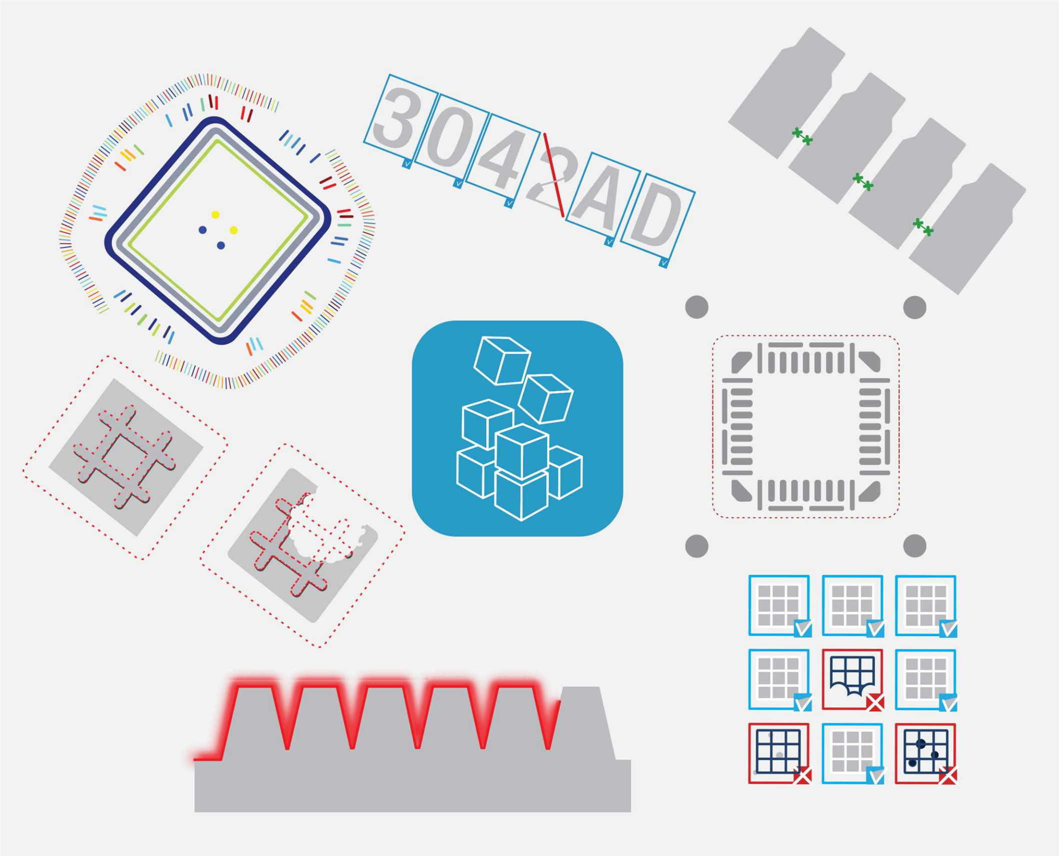
USER GUIDE

Open eVision Studio



© EURESYS S.A. 2024 - Doc. D120ET-Using Open eVision Studio-Open eVision-24.02.0.1198 built on 2024-03-26

This documentation is provided with **Open eVision Studio 24.02.0** (doc build **1198**).
www.euresys.com

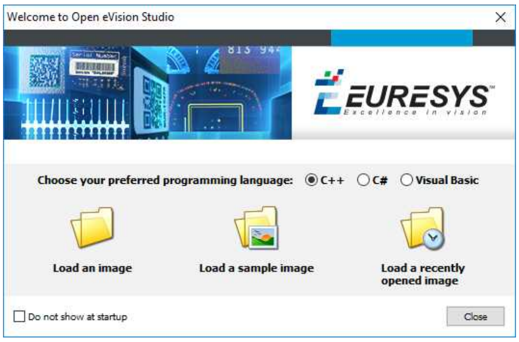
This documentation is subject to the General Terms and Conditions stated on the website of **EURESYS S.A.** and available on the webpage https://www.euresys.com/en/Menu-Legal/Terms-conditions. The article 10 (Limitations of Liability and Disclaimers) and article 12 (Intellectual Property Rights) are more specifically applicable.

Contents

1. Using Open eVision Studio

1.1. Selecting your Programming Language

When you start **Open eVision Studio** for the first time, the following welcome screen is displayed:



Welcome to Open eVision Studio

Choose your preferred programming language: ● C++ ○ C# ○ Visual Basic

□ Do not show at startup

1. Select your programming language.

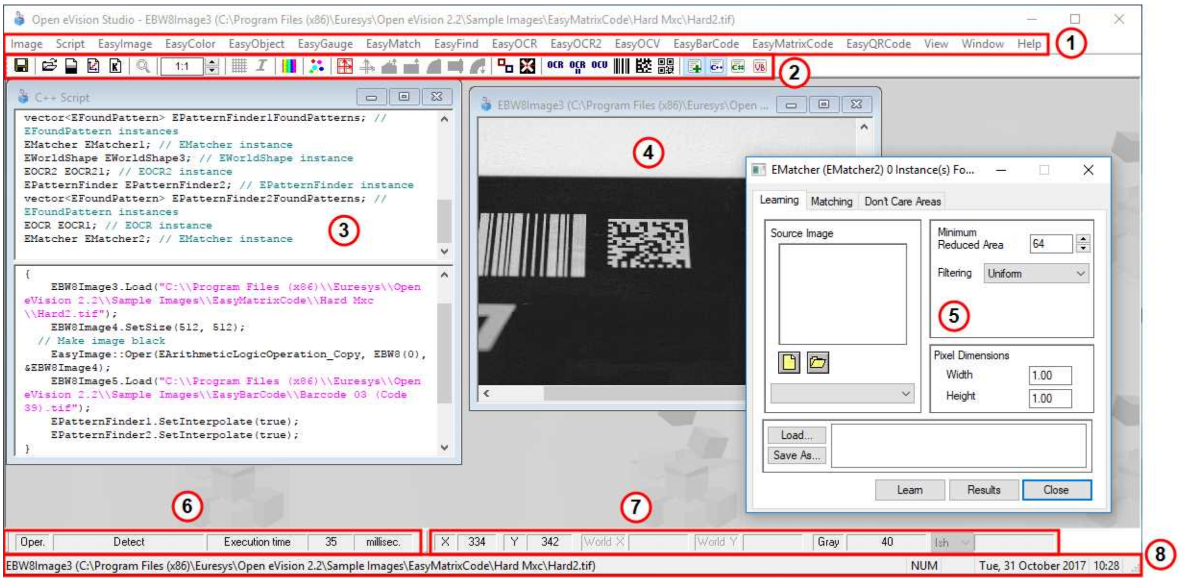
TIP
Your selection is saved and your programming language will be automatically selected next time you start **Open eVision Studio**.

NOTE
When you change your programming language, any script present in the scripting window is automatically deleted and the window content is reset.

2. Click on one of the `Load` buttons to already load one or several images for later processing.
3. Check the `Do not show at startup` box to hide this welcome screen next time you start **Open eVision Studio**.

TIP
To access this welcome screen at any time, and change this setting, go to the `Help` > `Welcome Screen` menu.

1.2. Navigating the Interface



```
vector<EFoundPattern> EPatternFinder1FoundPatterns; //
EFoundPattern instances
EMatcher EMatcher1; // EMatcher instance
EWorldShape EWorldShape3; // EWorldShape instance
EOCR2 EOCR21; // EOCR2 instance
EPatternFinder EPatternFinder2; // EPatternFinder instance
vector<EFoundPattern> EPatternFinder2FoundPatterns; //
EFoundPattern instances
EOCR EOCR1; // EOCR instance
EMatcher EMatcher2; // EMatcher instance
```

```
{
    EBW8Image3.Load("C:\\Program Files (x86)\\Euresys\\Open
eVision 2.2\\Sample Images\\EasyMatrixCode\\Hard Mxc
\\Hard2.tif");
    EBW8Image4.SetSize(512, 512);
  // Make image black
    EasyImage::Oper(EArithmeticLogicOperation_Copy, EBW8(0),
&EBW8Image4);
    EBW8Image5.Load("C:\\Program Files (x86)\\Euresys\\Open
eVision 2.2\\Sample Images\\EasyBarCode\\Barcode 03 (Code
39).tif");
    EPatternFinder1.SetInterpolate(true);
    EPatternFinder2.SetInterpolate(true);
}
```

Source Image

Minimum Reduced Area: 64

Filtering: Uniform

Pixel Dimensions

Width: 1.00

Height: 1.00

Oper.	Detect	Execution time	35	millisec.

X	334	Y	342	World X		World Y		Gray	40	Ish

EBW8Image3 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyMatrixCode\Hard Mxc\Hard2.tif)   NUM   Tue, 31 October 2017   10:28

Open eVision Studio graphical user interface (GUI) is organized as follows:

1. The *main menu bar* gives you access to the functions and tools of all libraries.

> TIP
>
> **Open eVision Studio** does not require any license and allows you to test all libraries. Of course, if you copy code from **Open eVision Studio** in your own application but you do not have the required license, you will receive a "missing license" error at run-time.

2. The *main toolbar* gives you quick access to main **Open eVision** objects such as images, shapes, gauges, bar codes, matrix codes...
3. The *script window* displays the code, in the programming language you selected, corresponding to the actions you perform in **Open eVision Studio**. You can save or copy this code in your own application at any time.
4. The *image windows* display the open images that you can process using the libraries and tools.
5. The *tool windows* enable you to easily configure all the available tools. The corresponding settings are automatically added in the script window for easy reuse.

> TIP
>
> Most tool windows are floating and you can easily move them outside the **Open eVision Studio** main window to make better use of your screen size.

6. The *execution time bar* displays the precise time taken for the execution of the selected functions (measured in milliseconds or microseconds) on your computer. This accurate measurement helps you to evaluate the performance of your application.

7. The *color toolbar* displays current information such as the X and Y coordinates of the cursor on an image and the corresponding pixel value.
8. The *status bar* displays general information about the application such as the active image file path...

1.3. Running Tools on Images

Step 1: Selecting a Tool

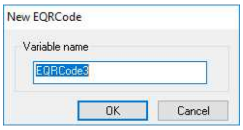
When you use **Open eVision Studio**, the first step is to select the library and the tool you want to use on your image.

To do so:

1. In the main menu bar, click on the library you want to use.
2. Click on the tool you want to use.

> TIP
> All libraries (except EasyImage, EasyColor and EasyGauge) expose only one tool named `New Xxx Tool`. Some of these libraries also expose additional functions.

3. In the dialog box, enter a `Variable name` for the variable that is automatically created and that will contain the result of the processing.



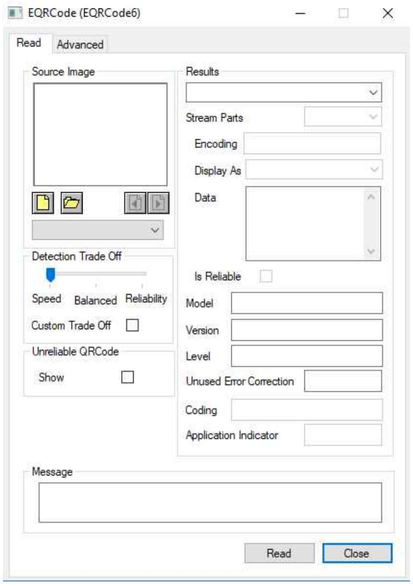
Variable name

EQRCode3

Example of variable creation dialog box for EasyQRCode

4. Click `OK`.

The selected tool dialog box opens.



Example of variable creation dialog box for EasyQRCode

The next step is "Step 2: Opening an Image" on page 8.

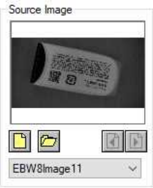
Step 2: Opening an Image

Once you have selected your library and your tool, you need to open an image to apply this tool.

In the `Source Image` area of the selected tool dialog box:

1. Open an image:
   - Click on the `Open an Image` button and select one or several (using SHIFT and CTRL) images on your computer.
   - Or select one of the images (or one of the ROIs, if any) already open in the drop-down list.

> NOTE
> You can select only images with an appropriate file format (JPG, PNG, TIFF or BMP) and in 8- and/or 24-bit depending on the library.



2. If you selected several images, activate one with the `Load Previous` or `Load Next` buttons.

The tool is automatically applied on any loaded image and, at this stage, the result is displayed based on the tool default settings.

The next step is "Step 3: Managing ROIs" on page 9.

Step 3: Managing ROIs

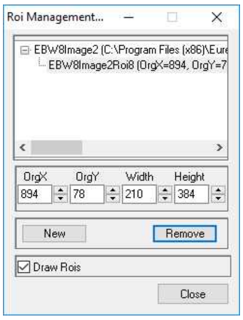
In some cases, most often to decrease the processing time or to single-out the object you want to read, you do not want to process the whole image but only one or several well defined rectangular parts of this image, or ROIs (Regions Of Interest).

> TIP
>
> In **Open eVision**, ROIs are attached to an image and exist only as long as the parent image is available.

Creating a ROI

1. Open the image:
   - □ If the image is already open, activate the corresponding image window.
   - □ If the image is not open yet, go to the main menu: `Image` > `Open...` to open one.
2. To create an ROI, go to the main menu: `Image` > `ROI Management...`.

The `ROI Management` window is displayed as illustrated below.



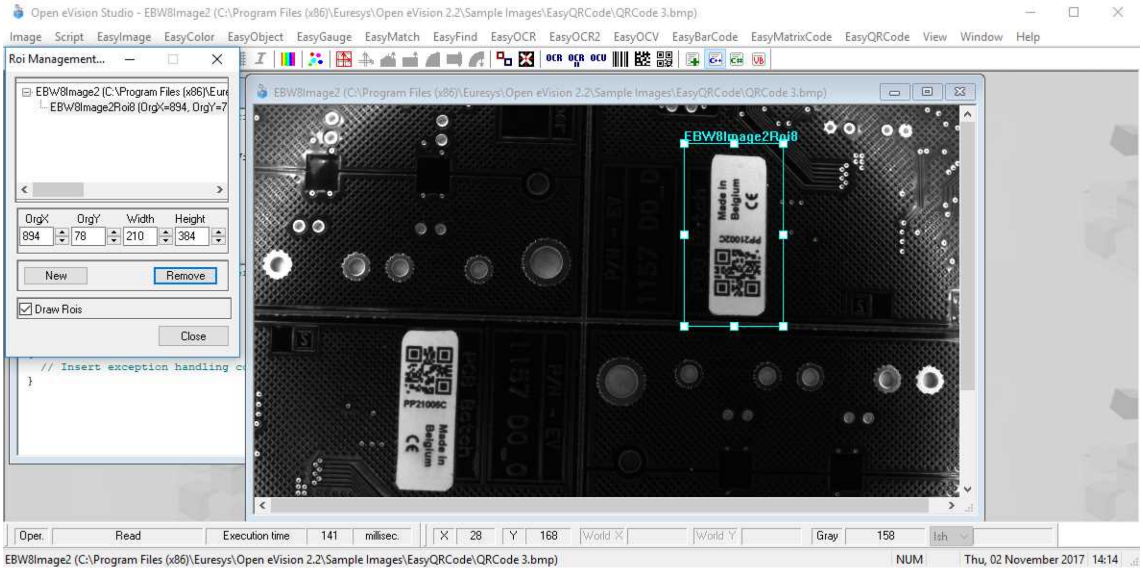
EBW8Image2 (C:\Program Files (x86)\Eure
EBW8Image2Roi8 (OrgX=894, OrgY=7

OrgX	OrgY	Width	Height
894	78	210	384

☑ Draw Rois

3. Select the image in the tree.
4. Click on the `New` button.
5. In the dialog box, enter a `Variable name` for the new ROI.

The ROI is represented as a color rectangle on your image as illustrated below.


Open eVision Studio - EBW8Image2 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 3.bmp)
Image Script EasyImage EasyColor EasyObject EasyGauge EasyMatch EasyFind EasyOCR EasyOCR2 EasyOCV EasyBarCode EasyMatrixCode EasyQRCode View Window Help
Roi Management...
New
Remove
Close
EBW8Image2 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 3.bmp)
EBW8Image2Roi8
Oper. Read Execution time 141 millisec.
NUM
Thu, 02 November 2017 14:14

EBW8Image2 (C:\Program Files (x86)\Eure
EBW8Image2Roi8 (OrgX=894, OrgY=7

OrgX	OrgY	Width	Height
894	78	210	384

☑ Draw Rois

X 28 Y 168 Gray 158

EBW8Image2 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 3.bmp)

6. Drag the ROI corner and side handles to move it to the required position.
7. Click on the `Close` button to close the `ROI Management` window .

The next step is "Step 4: Configuring the Tool" on page 11.

Managing ROIs

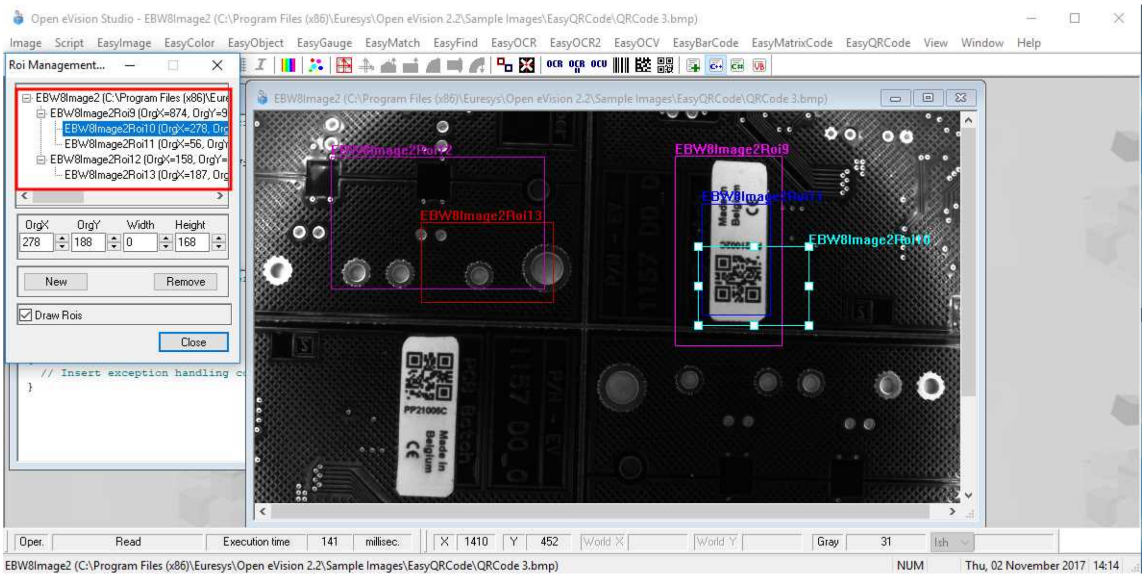
You can add, change and remove ROIs.

TIP
An image can have several ROIs. Each ROI can be attached directly to the image (meaning that its position is relative to the image) or to another ROI (meaning that its position is relative to this 'parent' ROI).

1. To manage ROIs, go to the main menu: `Image` > `ROI Management...`.

The `ROI Management` window is displayed with the ROI relation tree as illustrated below.

If the `Draw Rois` box is checked, all ROIs are displayed on the image with a different color.



Open eVision Studio - EBW8Image2 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 3.bmp)

Roi Management...

- EBW8Image2 (C:\Program Files (x86)\Eure
  - EBW8Image2Roi9 (OrgX=874, OrgY=9
    - EBW8Image2Roi10 (OrgX=278, Org
    - EBW8Image2Roi11 (OrgX=56, OrgY
  - EBW8Image2Roi12 (OrgX=158, OrgY=
    - EBW8Image2Roi13 (OrgX=187, Org

OrgX	OrgY	Width	Height
278	188	0	168

☑ Draw Rois

```
    // Insert exception handling c
}
```

Oper.	Read	Execution time	141	millisec.	X	1410	Y	452	World X		World Y		Gray	31

EBW8Image2 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 3.bmp) NUM Thu, 02 November 2017 14:14

2. Select an ROI in the ROI relation tree.
3. Drag the ROI corner and side handles to change the position and size of the selected ROI (as well as the position of all ROIs attached to it if any).
4. Click on the New button to add a new ROI attached to the selected ROI.

> TIP
> Select the image at the top of the ROI relation tree to attach the ROI directly to the image.

5. Click on the Remove button to delete the selected ROI (and all ROIs attached to it if any).
6. Click on the Close button to close the ROI Management window.

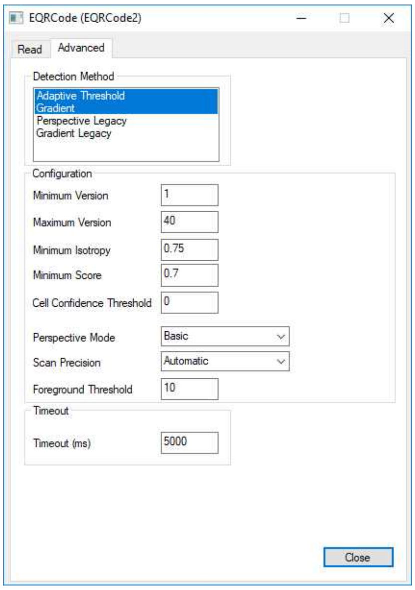
Step 4: Configuring the Tool

Once your image, including its ROIs if you created some, is ready, you need to configure your tool.

In the tool window:

1. Open the various tabs.

> TIP
> When you create a new tool, all parameters are set with their default value.



Detection Method
Adaptive Threshold
Gradient
Perspective Legacy
Gradient Legacy

Configuration
Minimum Version: 1
Maximum Version: 40
Minimum Isotropy: 0.75
Minimum Score: 0.7
Cell Confidence Threshold: 0
Perspective Mode: Basic
Scan Precision: Automatic
Foreground Threshold: 10

Timeout
Timeout (ms): 5000

Example of the parameter tab of an EasyQRCode tool

2. In each tab, set the value of the parameters as desired.

Please refer to the "Functional Guide" and to the "Reference Manual" for detailed information about the parameters, their function and their default value.

For specific actions such as learning or using gauges, please refer to the "Functional Guide".

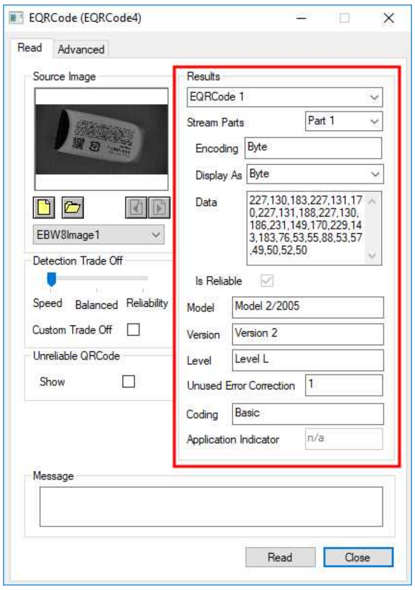
3. Run the tool and analyze the results as described in the next step "Step 5: Running the Tool and Checking Execution Time" on page 12.

Step 5: Running the Tool and Checking Execution Time

Once your tool parameters are set, run your tool and, if desired, check the execution time on your computer.

In the tool window:

1. Click on the `Read`, `Detect`, `Results` or `Execute` button (depending on the library function), to run the tool on the selected image.
2. Check the results on the image and in the Results field or area as illustrated below.



Results: EQRCode 1

Stream Parts: Part 1

Encoding: Byte

Display As: Byte

Data: 227,130,183,227,131,170,227,131,188,227,130,186,231,149,170,229,143,183,76,53,55,88,53,57,49,50,52,50

Is Reliable ☑

Model: Model 2/2005

Version: Version 2

Level: Level L

Unused Error Correction: 1

Coding: Basic

Application Indicator: n/a

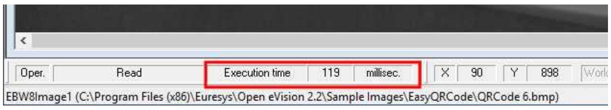
EBW8Image1

Custom Trade Off □

Show □

Example of results after reading a QRCode

3. If you do not have the expected results:
   - □ Try to change your parameters (start with default values then change one parameter at a time).
   - □ If your image is not good enough, try to enhance it as described in .
4. Check the execution time in the execution time bar at the bottom left of the main **Open eVision Studio** window.



```
Oper. | Read | Execution time | 119 | millisec. | X | 90 | Y | 898 | Worl
EBW8Image1 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 6.bmp)
```

The execution time

TIP

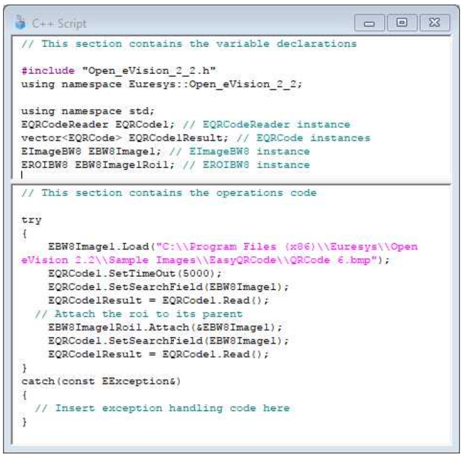
The execution time is the actual time that the processing took as measured on your computer. It depends your computer processor, memory, operating system... and, of course, on the processor load at the time of execution. Thus this execution time slightly varies from execution to execution.

5. To get a more representative execution time, click on the `Read`, `Detect`, `Results` or `Execute` button several times and calculate the mean execution time.
6. If your application requires that you reduce the execution time, try:
   - □ To change the tool parameters,
   - □ To add one or several ROIs on your image,
   - □ To enhance your image.

The next step is "Step 6: Using the Generated Code" on page 14.

Step 6: Using the Generated Code

By default, **Open eVision Studio** translates all the operations you perform in the interface into code in the language you selected as illustrated below.



```
// This section contains the variable declarations

#include "Open_eVision_2_2.h"
using namespace Euresys::Open_eVision_2_2;

using namespace std;
EQRCodeReader EQRCode1; // EQRCodeReader instance
vector<EQRCode> EQRCode1Result; // EQRCode instances
EImageBW8 EBW8Image1; // EImageBW8 instance
EROIBW8 EBW8Image1Roi1; // EROIBW8 instance
```

```
// This section contains the operations code

try
{
    EBW8Image1.Load("C:\\Program Files (x86)\\Euresys\\Open
eVision 2.2\\Sample Images\\EasyQRCode\\QRCode 6.bmp");
    EQRCode1.SetTimeOut(5000);
    EQRCode1.SetSearchField(EBW8Image1);
    EQRCode1Result = EQRCode1.Read();
  // Attach the roi to its parent
    EBW8Image1Roi1.Attach(&EBW8Image1);
    EQRCode1.SetSearchField(EBW8Image1);
    EQRCode1Result = EQRCode1.Read();
}
catch(const EException&)
{
  // Insert exception handling code here
}
```

Once your tool results suit you, you can save or copy this generated code to use it in your own application.

Copy and paste the code in your application

In the script window:

1. Select the code section you want to copy.
2. Right click on this code and click `Copy` in the menu.
3. Go to your development environment tool and paste the code in place.

Save the code

1. Go to the **Script** menu.
2. Click on `Save Script As...`.
3. Enter a file name and path to save the code as a text file.

Manage the generated code

In the **Script** menu, you can:

- Select the programming language (please note that if you change the language, the script window content is automatically deleted).
- Activate or deactivate the `Script Code Generation`. Deactivate this option if you want to perform some operations without saving them as code.

1.4. Pre-Processing and Saving Images

When should you pre-process your images?

Of course, the best situation is to set up your image acquisition system to have good and easy to process images so the **Open eVision** tools run smoothly and efficiently.

If this is not possible or easy to achieve, you can pre-process your images or your ROIs to enhance and prepare them for the **Open eVision** tool you want to run.

Using the various available functions, you can adjust the gain and offset of your image, apply a convolution, threshold, scale, rotate and white balance your image, enhance contours... using EasyImage and EasyColor functions.

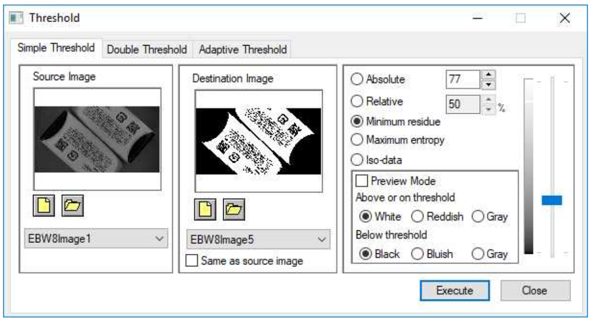
Pre-processing images

The difference between pre-processing an image and running tools is that the pre-processing generates a new image while the tools mainly extract and retrieve information from the image without changing it.

To pre-process an image or an ROI:

1. In the main menu bar, click on the library you want to use (EasyImage or EasyColor).
2. Click on the function you want to use.

Most function dialog boxes are similar to the one illustrated below with 2 image selection areas and a parameter setting area.



Threshold

Simple Threshold | Double Threshold | Adaptive Threshold

Source Image: EBW8Image1

Destination Image: EBW8Image5
☐ Same as source image

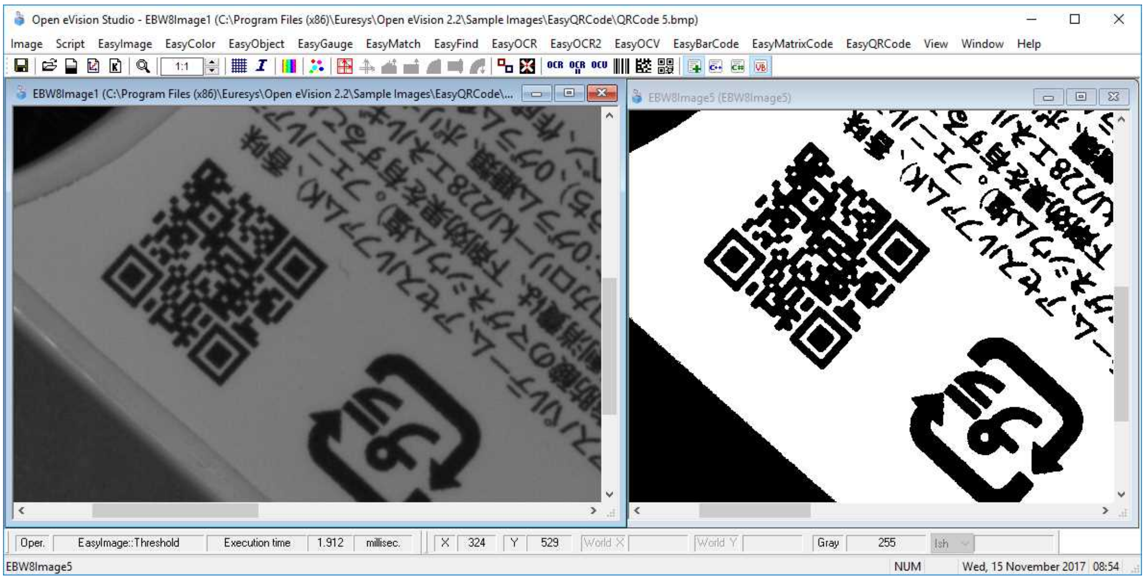
○ Absolute 77
○ Relative 50 %
● Minimum residue
○ Maximum entropy
○ Iso-data

☐ Preview Mode
Above or on threshold: ● White ○ Reddish ○ Gray
Below threshold: ● Black ○ Bluish ○ Gray

Example of a pre-processing dialog box (Threshold with EasyImage)

3. If there are multiple versions for your selected function, open the corresponding tab.
4. In the `Source Image` area, open the source image (as described in "Step 2: Opening an Image" on page 8).
5. In the `Destination Image` area, open or create a new destination image.
6. Set your parameters.
7. Click on the `Execute` button.

The pre-processed image is available in the destination image as illustrated below.


Open eVision Studio - EBW8Image1 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\QRCode 5.bmp)
Image Script EasyImage EasyColor EasyObject EasyGauge EasyMatch EasyFind EasyOCR EasyOCR2 EasyOCV EasyBarCode EasyMatrixCode EasyQRCode View Window Help
EBW8Image1 (C:\Program Files (x86)\Euresys\Open eVision 2.2\Sample Images\EasyQRCode\...
EBW8Image5 (EBW8Image5)
EBW8Image5
NUM
Wed, 15 November 2017 08:54

Oper. EasyImage::Threshold Execution time 1.912 millisec. X 324 Y 529 World X World Y Gray 255

Source and destinations images (Threshold with EasyImage)

8. If you want to use the destination image outside of **Open eVision Studio**, save it as described below.

Saving an image

1. Click on the image you want to save to makes its window active.
2. To open the save menu either:
   - □ Right-click in the image
   - □ Or open the main menu > `Image`
3. Click on `Save as...`.
4. Select the file format (JPEG, JPEG2000, PNG, TIFF or Bitmap).
5. Enter a name and select a path.
6. Click on the `Save` button.

2. Tutorials

2.1. Preliminaries

- About the application basic interface, operations and workflow, please refer to "Getting Started" on page 1, available through the **Help** menu. About the **Open eVision** functional concepts, and about the API classes, methods, properties, refer to the main section "Using Open eVision" on page 1.
- There are often many different ways to get the same results through the graphical interface. The tutorials show only one way for each case.
- All the images used in the tutorials are located by default in the following subdirectory. As this root path is constant, it is not repeated in each tutorial.

  ```
  C:\Users\Public\Documents\Euresys\Open eVision xx.xx\Sample Images
  ```
- By default, in the main toolbar, the **Activate/Deactivate script code generation** toggle button is on. This will insert automatically into the script window the code corresponding to the actions that are taken through the graphical interface. But note that the automatic operations are not inserted into the script until they have been explicitly performed by clicking a dialog box button (**Execute**, **Find**, **Learn**...).

2.2. EasyImage

Converting a Gray-Level Image into a Binary Image

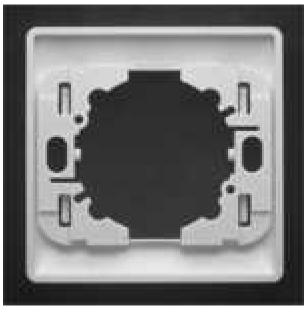
Thresholding

Single Thresholding - Double Thresholding - Histogram-Based Single Thresholding - Histogram-Based Double Thresholding

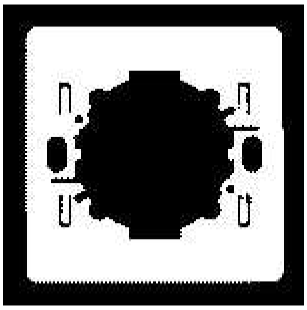
Objective

Following this tutorial, you will learn how to use EasyImage to convert a gray-level source image into a binary destination image. Thresholding an image transforms all the gray pixels into black or white pixels, depending on whether they are below or above a specified threshold. Thresholding an image makes further analysis easier.

You'll need first to load an image (step 1). Then you'll set the thresholding parameters (step 2), and perform the conversion (step 3).



Gray-level source image



Black and white destination image, after thresholding

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Threshold`.
2. In the `Simple Threshold` tab, click the `Open` icon of the Source Image area, and load the image file `EasyMatch\Switch1.tif`.
3. Keep the default variable name for the new Image object, and click `OK`.

Step 2: Set the thresholding parameters

1. In the right area of the `Threshold` dialog box, move the slider to change the threshold, and see directly in the source image a preview of the result.
2. Select the `Minimum residue` option to set a pre-defined algorithm that finds automatically the right threshold.

Step 3: Perform the conversion

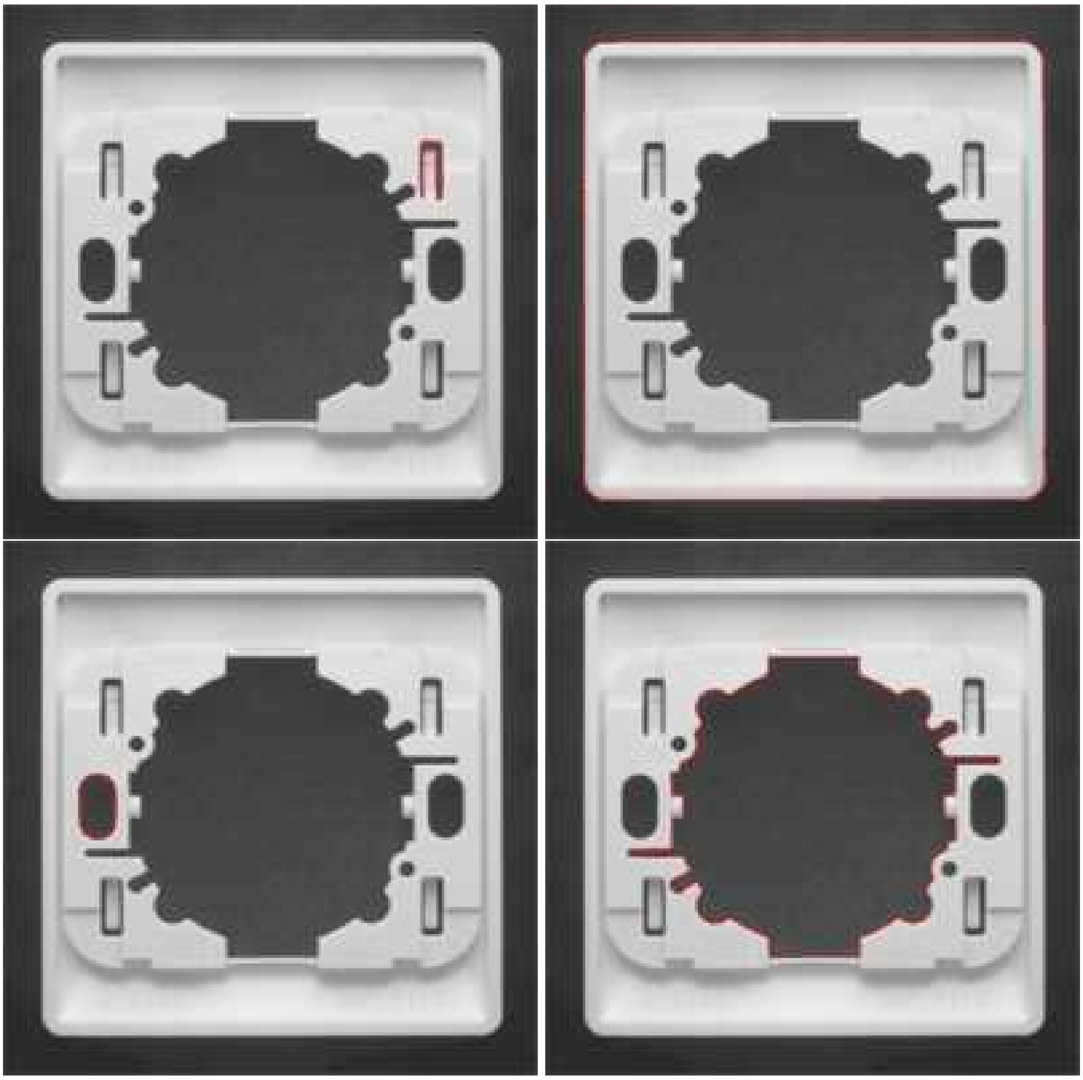
1. Click the `New` icon in the `Destination Image` area to create a new destination image.
2. Keep the default settings for the new Image object, and click `OK`.
3. In the `Threshold` dialog box, click `Execute` to perform the thresholding in the destination image.

Extracting an Object Contour

Objective

Following this tutorial, you will learn how to use EasyImage to trace an object outline in a gray-level image. The contour extraction allows you to get in a path vector all the points that constitute an object contour, just by clicking an edge of this object.

You'll need first to load an image (step 1) and set a vector that will contain all the contour points (step 2). Then you'll click an object edge, and the contour will be extracted automatically (step 3).



Contours are extracted from object edges

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Contour`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyMatch\Switch1.tif`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Set the destination vector

1. Click the `New` icon in the Destination Vector area.

2. Keep the default settings and variable name for the new vector object.
3. Click OK.

Step 3: Extract the contour

1. When moving the cursor above the image, an arrow appears.
2. Move the arrow above an object edge, and click.

From this object edge, a contour is traced, and a red line appears around the object.

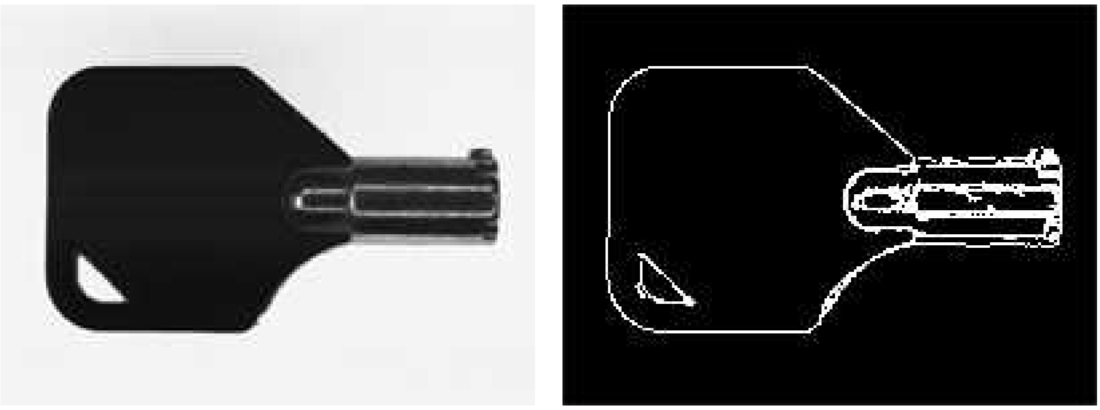
The destination vector is filled with the points constituting the contour.

Transforming a Gray-Level image into its Black and White Edges

Objective

Following this tutorial, you will learn how to use EasyImage to transform a gray-level image to a binary image, keeping only the edges detected in the image. The conversion uses the Canny edge detector algorithm.

You'll need to load a source image (step 1), and simply apply the Canny edge transformation (step 2).



Source image (left) and destination image after Canny edge transformation (right)

Step 1: Load the source image

1. From the main menu, click EasyImage, then Canny Edge Detector.
2. Keep the default variable name, and click OK.
3. Click the Open icon of the Source Image area, and load the image file EasyImage\Key1.tif.
4. Keep the default variable name, and click OK.

Step 2: Apply the Canny edge transformation

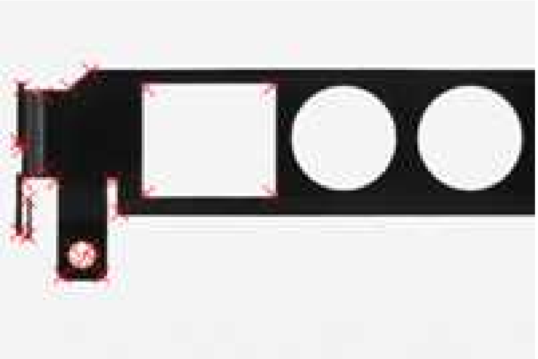
1. Click the `New` icon in the Destination Image area to create a new destination image.
2. Keep the default settings, and click `OK`.
3. In the canny edge detector dialog box, click `Apply` to perform the operation in the destination image.

Detecting the Corners of an Object Using Harris Corner Detector

Objective

Following this tutorial, you will learn how to use EasyImage to detect the corners of an object. The detection uses the Harris corner detector algorithm.

You'll need to load a source image (step 1), and simply apply the Harris corner detection (step 2).



Corners are detected in the source image

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Harris Corner Detector`.
2. Keep the default variable name, and click `OK`.
3. Click the `Open` icon of the Source Image area, and load the image file `EasyGauge\Bracket1.tif`.
4. Keep the default variable name, and click `OK`.

Step 2: Apply the Harris corner detection

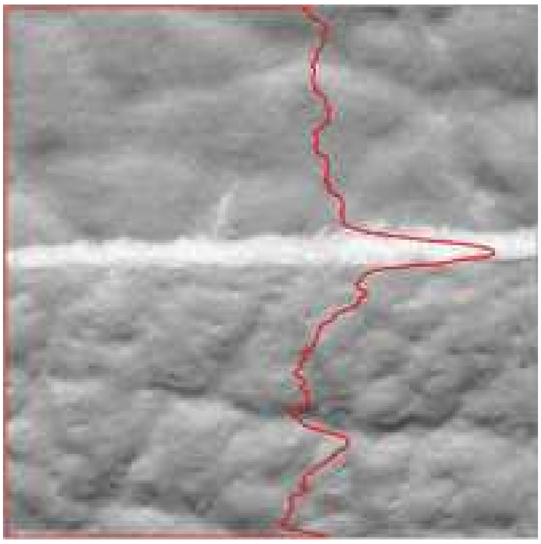
1. In the Harris corner detector dialog box, enter 2.3 for the Scale property.
2. Click `Apply` to perform corners detection.
3. Click `Results` to display the coordinates of all detected corners.
4. The `Columns` button allows you to display additional properties in the results list.

Detecting a Horizontal or Vertical Line Using Projection

Objective

Following this tutorial, you will learn how to use EasyImage to detect defects (horizontal/vertical line) in a gray-level image.

You'll need first to load a source image (step 1), set a vector (step 2), and then detect the defect (horizontal line) (step 3).



Defects can be detected using the image projection

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Projection`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyImage\Leather.bmp`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Set the destination vector

1. Click the `New` icon in the Destination Vector area.
2. Select the BW32 option for the vector type, and click `OK`.

Step 3: Detect the defects

1. In the Image Projection dialog box, select the `column` button, and click `Execute` to perform the operation.
2. The resulting vector and the corresponding plot are displayed in the destination vector window. The graphical result also appears on the image. Each vector value is the sum of all pixels values across the corresponding horizontal row (or vertical column). By this mean, horizontal (or vertical) defects are easily detected.

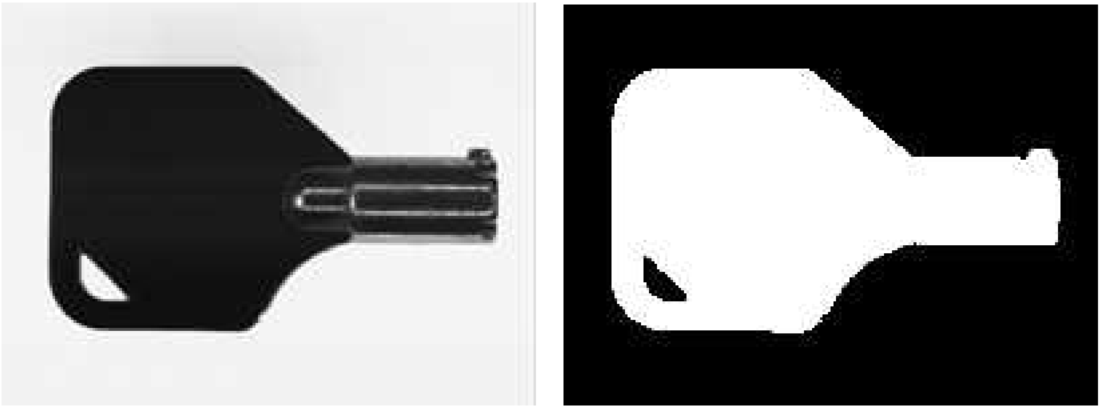
Creating a Flexible Mask

Using Flexible Masks

Objective

Following this tutorial, you will learn how to create a flexible mask from a source image, to restrict a future processing to an arbitrary-shaped do-care area.

Flexible masks can be created in any ways to build a bi-level image. Here, we will first load the source image (step 1), and then successively invert it, and threshold it (steps 2-3). The resulting image —the flexible mask— will be saved as a new image file (step 4). This new image file is a bi-level image. However, there are still black areas that need to be erased, before using the image as a flexible mask. You can use a third-party software, such as Paint, to clear the unwanted areas.



Source image (left) and flexible mask image (right)

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Arithmetic & logic`.
2. Click the `Open` icon of the Source Image 0 area, and load the image file `EasyImage\Key1.tif`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Invert the image

1. Click the `New` icon of the `Destination` area.
2. Keep the default settings for the new image object, and click `OK`.
3. In the `Operation` drop-down list, select `Invert`, and click `Execute`

Step 3: Threshold the image

1. From the main menu, click `EasyImage`, then `Threshold`.
2. In the Source Image area, select the inverted image from the drop-down list.
3. Click the `New` icon of the Destination area.
4. Keep the default settings for the new image object, and click `OK`.

5. Select the Absolute option, enter '46' as the threshold value, and click `Execute`.

Step 4: Save the flexible mask

1. Right-click in the destination image, and select `Save As...`.
2. Type a file name for the new flexible mask file. Finally, click `Save`.

> TIP
>
> The new image now is a bi-level image. However, there are still black areas that need to be erased, before using the image as a flexible mask. You can use a third-party software, such as Paint, to clear the unwanted areas.

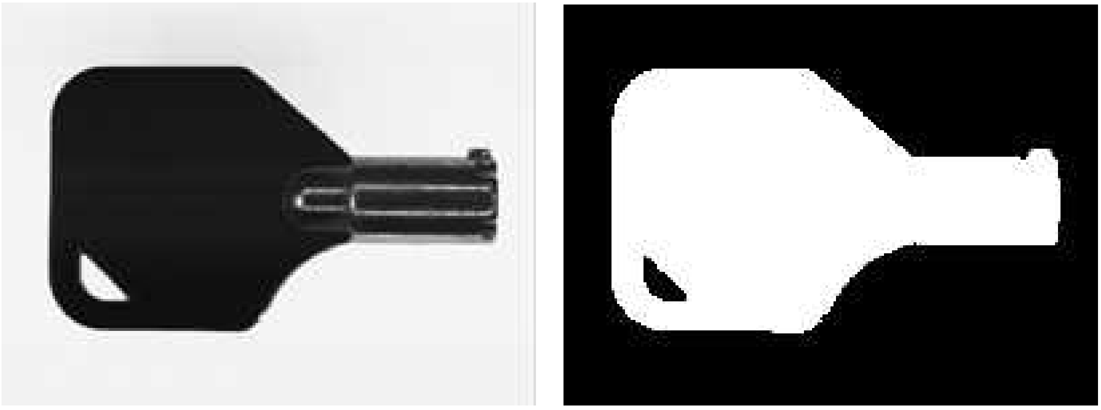
Computing Gray-Level Statistics Using a Flexible Mask

Using Flexible Masks

Objective

Following this tutorial, you will learn how to compute gray-level statistics on an arbitrary-shaped area only.

You'll need first to load a source image (step 1), and a flexible mask image (step 2). The mask image must be applied on the source image (step 3), to separate do-care areas (that must be considered) and don't-care areas (that should not be considered). Finally, the gray-level statistics are computed on the do-care area only (step 4).



Source image (left) and flexible mask image (right)

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Image Statistics`, `Gray Scale`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyImage\Key1.tif`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Load the flexible mask image

1. From the main menu, click `Image`, then `Open...`.
2. Load the image file `EasyImage\Mask2.bmp`.
3. Keep the default variable name for the new image, and click `OK`.

Step 3: Apply the flexible mask on the source image

1. In the Mask area of the `Gray Scale Image Pixels Statistics` dialog box, select the mask image from the drop-down list.
2. The source image preview in the dialog box shows (in red diagonal lines) the don't-care area, that is the area that will be not be considered when computing the gray-level statistics.

Step 4: Compute the gray-level statistics

1. Select the `Pixel Count` check-box.
2. Click `Execute`.

The results are displayed in the read-only fields below.

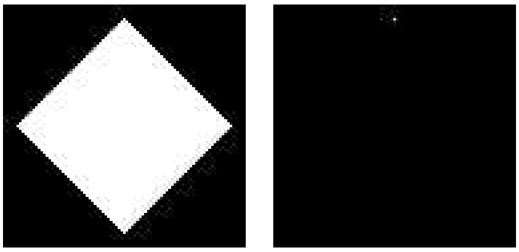
Detecting the Corners of an Object Using Hit-and-Miss Transform

Hit-and-Miss Transform

Objective

Following this tutorial, you will learn how to use EasyImage to detect top corners in an image, using the hit-and-miss transform.

You'll need to load a source image (step 1), set the kernel that represents a top corner (step 2), and then set a destination image and simply execute the hit-and-miss transform (step 3).



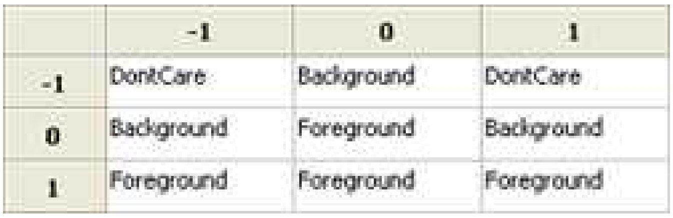
Source image (left) and top corner detected in the source image (white dot)

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Hit And Miss`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyImage\Diamond.bmp`.
3. Keep the default variable name, and click `OK`.

Step 2: Set the hit-and-miss kernel

- In the Hit And Miss dialog box, set the kernel according to the following values:



	-1	0	1
-1	DontCare	Background	DontCare
0	Background	Foreground	Background
1	Foreground	Foreground	Foreground

Kernel that detects top corners

Step 3: Apply the hit-and-miss transform

1. Click the New icon of the Destination Image area.
2. Keep the default parameters and variable name, and click OK.
3. Click Execute to perform the operation.

The top corner (white dot) is detected.

4. Try with other kernel configurations to detect the other corners.

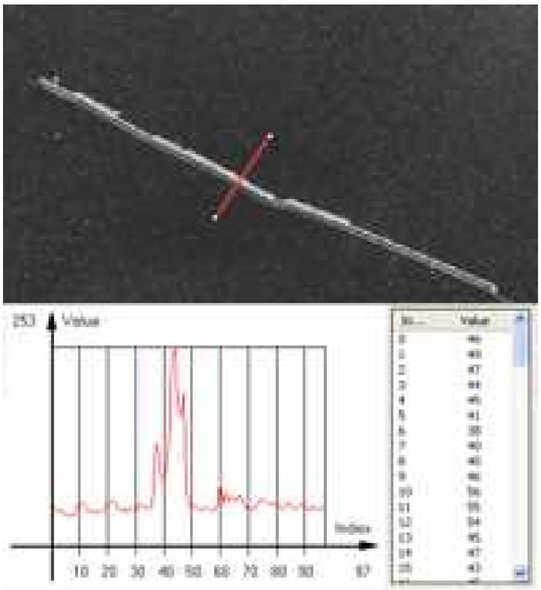
Extracting a Vector Using Profile Function

Profile Sampling

Objective

Following this tutorial, you will learn how to use EasyImage to detect scratches.

You'll need first to load an image (step 1), set a destination vector, and detect the scratches (step 2).



Scratches can be detected using a profile

Step 1: Load the source image

From the main menu, click `EasyImage`, then `Profile`.

1. Click the `Open` icon of the `Source Image` area, and load the image file `EasyImage\Plastic.tif`.
2. Keep the default variable name for the new image object, and click `OK`.

Step 2: Set the destination vector and detecting the scratches

1. Click the `New` icon in the Destination Vector area.
2. Select the BW8 option for the vector type, and click `OK`.
3. A profile appears on the image (red line segment). In the destination vector window, vector values correspond to pixels along the line segment.

The scratch is detected as a sharp deviation in the vector graph.

4. Using the mouse, drag the handles to move or resize the red line segment, and observe the plot evolution.

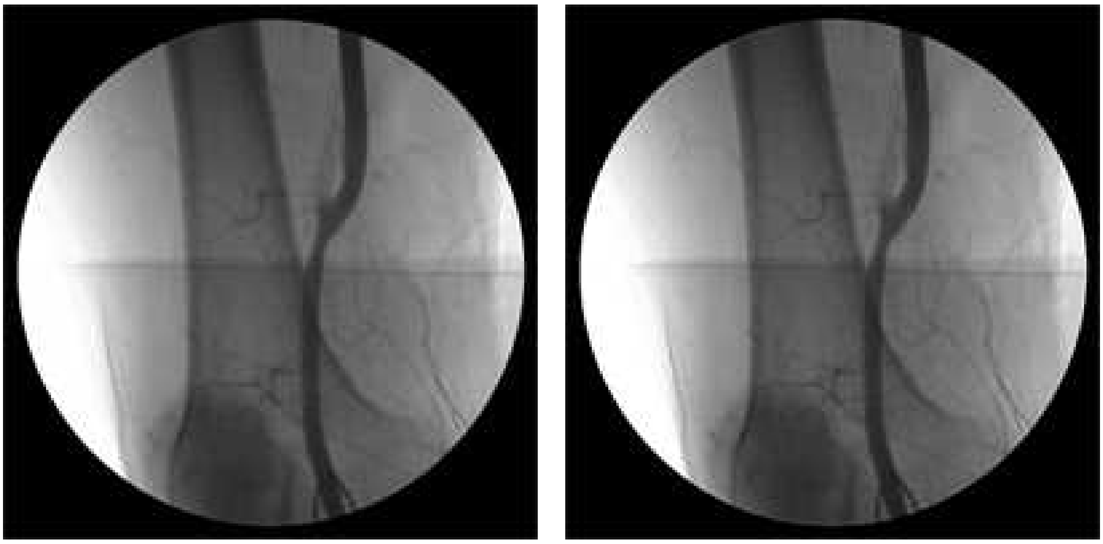
The sharp deviation appears whenever the line segment is placed across the scratch.

Enhancing an X-ray image

Objective

Following this tutorial, you will learn how to use EasyImage to enhance an X-ray image.

You'll need first to load an image (step 1), then define convolution parameters to enhance the image (step 2).



Source image (left) and enhanced image, after predefined and user-defined convolutions (right)

Step 1: Load the source image

1. From the main menu, click EasyImage, then Convolution.
2. Click the Open icon of the Source Image area, and load the image file EasyImage\XRay.bmp.
3. Keep the default variable name for the new image object, and click OK.
4. Click the New icon of the Destination Image area to create a new destination image.
5. Keep the default variable name and click OK.

Step 2: Set the convolution parameters

1. From the Predefined kernels drop-down list, select Highpass2, and click Execute to perform the operation.

The image is no longer blurred but the result is bad because the filter has revealed the noise of the source image. We need to create a new convolution kernel that will apply a softer high-pass filtering.

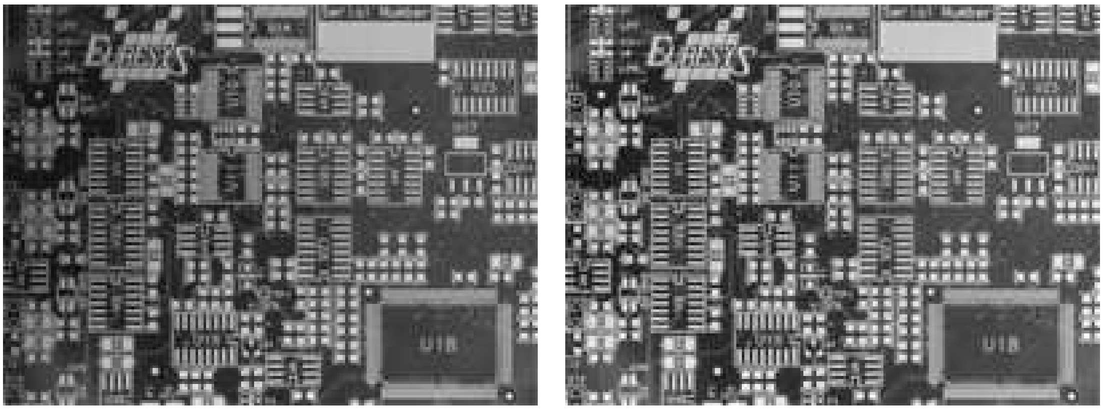
2. Click the New icon next to the User defined kernels drop-down list.
3. Keep the default dimension (3x3) and variable name, and click OK.
4. Enter the following kernel data from left to right and top to bottom: -1, -1, -1; -1, 15, -1; -1, -1, -1, and click Apply.
5. Click Execute in the Convolution dialog box to perform the operation. The image is much clearer now.

Correcting Non-Uniform Illumination

Objective

Following this tutorial, you will learn how to use EasyImage to correct non-uniform illumination in an image.

You'll need first to load an image (step 1), load a light reference image (step 2), and perform the correction (step 3).



Source image, with non-uniform illumination (left) and corrected image (right)

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Uniformize`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyImage\Board (original).tif`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Load the reference image

1. Click the `Open` icon of the `Light Reference` area, and load the image file `EasyImage\Board (light reference).tif`.

To obtain the light reference image, we used a white screen illuminated in the same condition as the board (original image).

2. Keep the default variable name for the new image object, and click `OK`.

Step 3: Perform the correction

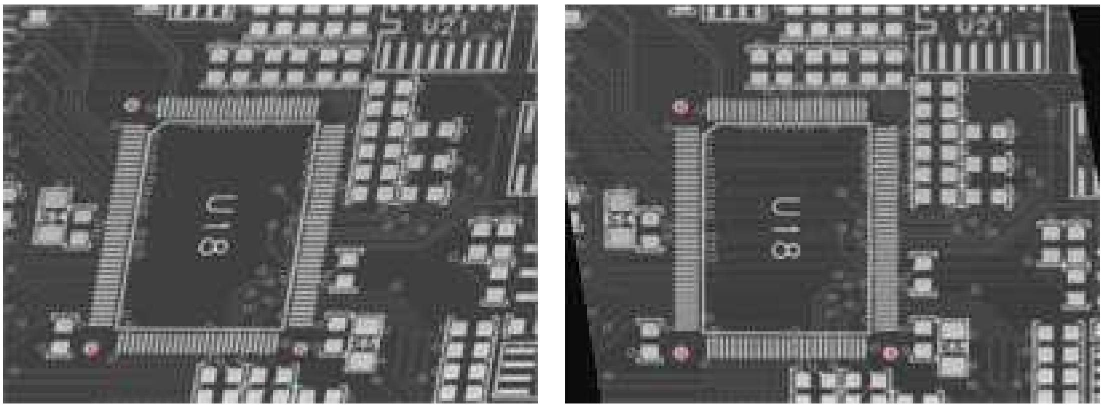
1. Click the `New` icon in the `Destination Image` area to create a new destination image.
2. Keep the default values and click `OK`.
3. Click `Execute` to perform the operation.
4. In both source and destination images, right-click and select `3D Rendering`.
5. In the new 3D windows, click and drag the mouse to rotate the view. Compare the profiles.

Correcting Shear Effect

Objective

Following this tutorial, you will learn how to use EasyImage to correct a shear effect in an image. The following image is taken by a line-scan camera. The camera sensor was misaligned, resulting in a so-called shear effect.

You'll need first to load an image (step 1), create a destination image (step 2), and then set pivots parameters to perform the correction (step 3).



Source image, with a shear effect (left) and corrected image (right)

Step 1: Load the source image

1. From the main menu, click `EasyImage`, then `Register`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyImage\Shear.tif`.
3. Keep the default variable name for the new image object, and click `OK`. Three pivots points appear in the image.

Step 2: Create a destination image

1. Click the `New` icon of the `Destination Image` area.
2. Enter '768' and '576' as image width and height, and click `OK` to accept the default name. Three pivots points appear in the image.

Step 3: Set the pivots parameters

1. In the source image, using the mouse, drag each pivot to the center of the fiducial marks (the dots around the U18 area).

Notice that the source pivots coordinates, in the Register dialog box, have changed accordingly.

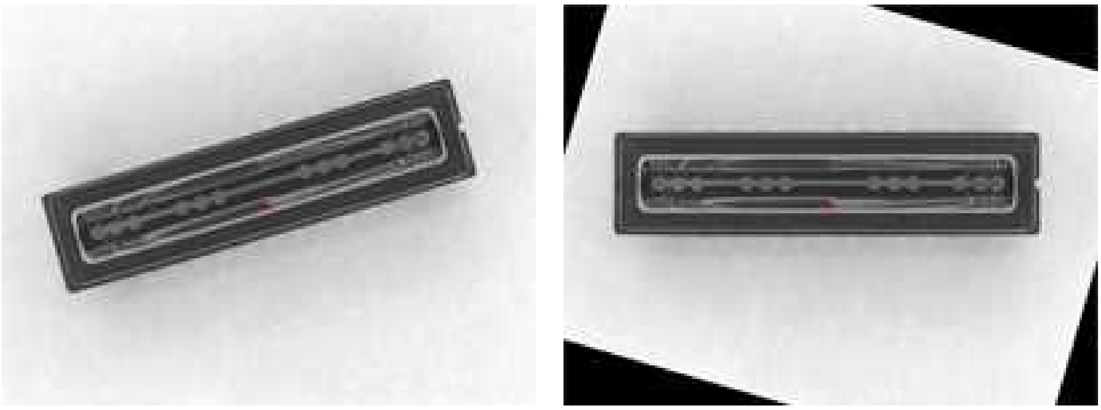
2. To correct the image, enter the following destination pivots coordinates:
   - X0: 170
   - Y0: 495
   - X1: 470
   - Y1: 495
   - X2: 170
   - Y2: 144
3. Click `Execute` to perform the operation.

Correcting Skew Effect

Objective

Following this tutorial, you will learn how to use EasyImage to correct skew effect in an image.

You'll need first to load an image (step 1), create a destination image (step 2), and then set a correction angle to perform the correction (step 3).



Source image, with a skew effect (left) and corrected image (right)

Step 1: Loading the source image

1. From the main menu, click `EasyImage`, then `Scale and Rotate`.
2. Click the `Open` icon of the `Source Image` area, and load the image file `EasyImage\CCD.tif`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Creating a destination image

1. Click the `New` icon of the `Destination Image` area.
2. Enter '768' and '576' as image width and height, and click `OK` to accept the default name.

Step 3: Setting the correction angle

1. Select the `Rotate` option, and enter -16.17 in the Angle (Deg) field. (To measure this rotation angle, refer to Measuring the rotation angle of an object.)
2. From the Interpolation bits drop-down list, select 8 bits to get a better result.
3. Click `Execute` to perform the operation.

2.3. EasyColor

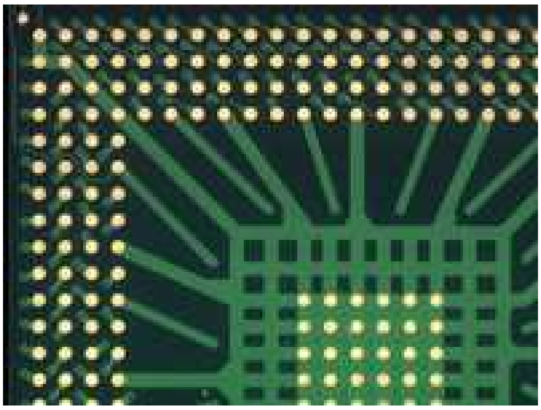
Performing Thresholding on Color Images

Color Components

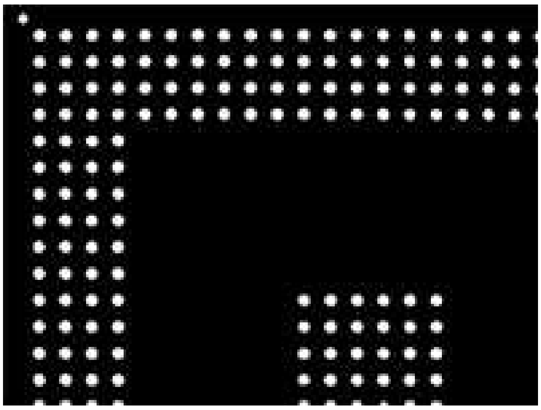
Objective

Following this tutorial, you will learn how to use EasyColor to segment a color source image, by setting a threshold value for each color component of the current color system. For example, to retrieve the solder pads on a PCB, you'll perform a color segmentation based on the golden pixels (H), with a loose discrimination on the brightness (L) and saturation (S), to eliminate surface and lighting effects.

You'll need first to load a color source image, create a destination image, and a color lookup table (steps 1-3). Then, you'll set the color system and tune each component tolerance to get a satisfying segmentation of the solder pads (step 4).



Source image



Thresholded image

Step 1: Load the source image

1. From the main menu, click `EasyColor`, then `Threshold`.
2. Click the `Open` icon of the `Source Image` area, and load the image file `EasyColor\BGA Substrate Color.jpg`.
3. Keep the default variable name for the new Image object, and click `OK`.
4. Disable the `Preview Mode` check-box to see the raw source image.

Step 2: Create a destination image

1. Click the New icon of the Destination Image area.
2. Keep the default variable name for the new Image object, and click OK.

Step 3: Create a color lookup table

1. Click the New icon of the Color Lookup area.
2. Keep the default variable name for the new color lookup object, and click OK.

Step 4: Perform the color segmentation

1. Select LSH from the Color System drop-down list.
2. In the source image, click in a golden pad. The pixel lightness, saturation and hue values are updated in the Color Threshold dialog box.
3. Adjust the tolerance of lightness and saturation to enlarge the range of thresholded pixels, until you get a satisfying segmentation in the destination image.

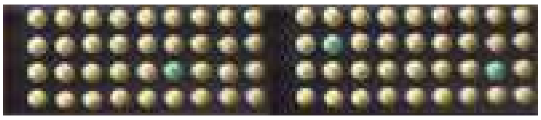
Performing Color Segmentation

Color Components

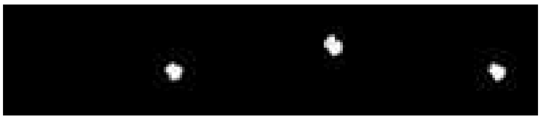
Objective

Following this tutorial, you will learn how to use EasyImage to perform color segmentation.

You'll need first to load an image (step 1), create a color look-up table (step 2), and perform the segmentation (step 3).



Source image



Segmented image

Step 1: Load the source image

1. From the main menu, click EasyColor, then Threshold.
2. Click the Open icon of the Source Image area, and load the image file EasyColor\Pills.tif.
3. Keep the default variable name for the new image object, and click OK.
4. Disable the Preview Mode check-box to see the raw source image.

Step 2: Create a color lookup table

1. Click the `New` icon of the Color Lookup area.
2. Keep the default variable name for the new color lookup object, and click `OK`.

Step 3: Perform the color segmentation

1. Select `LSH` from the `Color System` drop-down list.
2. In the source image, click in the center of a green pill. The pixel lightness, saturation and hue values are updated in the `Color Threshold` dialog box.
3. Increase the lightness tolerance up to 120. Increase the saturation tolerance up to 50.
4. Enable the `Preview Mode` check-box to see the result of the segmentation. If needed, click in the green pills to improve the result.
5. Click the `New` icon of the Destination Image area.
6. Keep the default settings for the new Image object, and click `OK`.
7. The new image is automatically thresholded. Clicking `Execute` will insert the corresponding code into the script windows.

2.4. EasyObject

Removing Non-Significant Objects After Image Segmentation

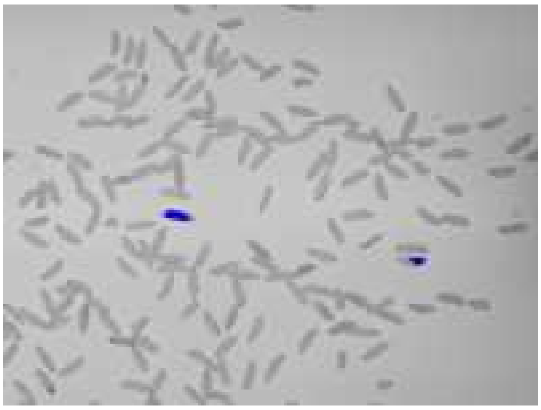
Image Segmenter

Objective

Following this tutorial, you will learn how to use EasyObject to detect bad rice grains (largely dark) among many normal rice grains (largely light).

You'll need first to load the source image (step 1). Then you'll perform the image segmentation, based on a threshold value (step 2). All the detected objects are dark, but some are too small to be significant. So, you'll set a minimum object area (number of pixels), and remove the smallest objects (step 3). Finally, you'll get only the objects that really represent bad rice grains.

Source image



Bad rice grains are detected

Step 1: Load the source image

1. From the main menu, click `EasyObject`, then `New EasyObject Tool`.
2. Keep the default variable name for the new object, and click `OK`.
3. In the `Encoder` tab, click the `Open` icon of the Source Image area, and load the image file `EasyObject\Rice.jpg`.
4. Keep the default variable name for the new image, and click `OK`.

Step 2: Perform image segmentation

1. In the `Encoder` tab, select the `Black Layer` check-box, and unselect the `White Layer` check-box.
2. Click the `...` button around the `Threshold` field. In the Threshold dialog box, select `Absolute`, enter '115', and click `OK`.
3. Click `Encode` to detect the dark objects. In the image, each object is drawn using a different color.
4. Click `Results` to display the list of all the detected objects. Clicking an object in the image highlights it in the list, and vice versa.

Step 3: Remove the smallest objects

1. In the objects list, click `Columns`.
2. Tick the `Area` check-box, and click `OK`. In the list view, a new Area column appears, displaying each object area.
3. Click the Area column header to sort the objects. There are many small objects (area < 100) that may be considered as noise.
4. In the `Selection` tab, select `Area` from the Feature drop-down list. Select '`Less`' from the Mode drop-down list. In the Threshold field, enter '100'.
5. Click `Remove`. All the objects with an area smaller than 100 pixels have been removed from the list. The two remaining objects are the bad rice grains.

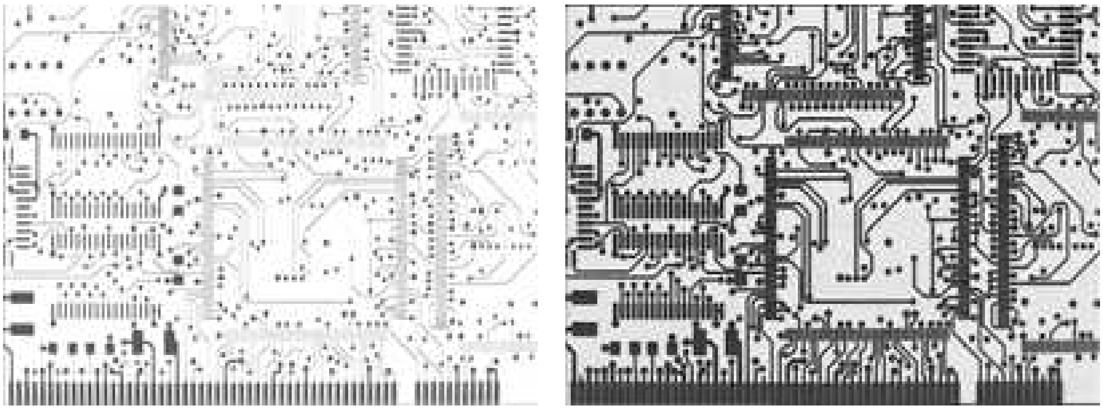
Detecting Differences Between Images Using Min-Max References

Selecting and Sorting Blobs

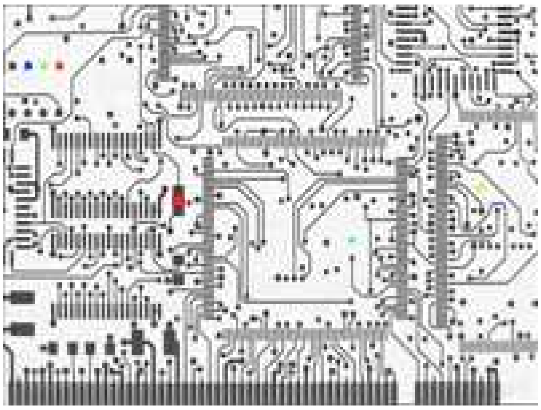
Objective

Following this tutorial, you will learn how to use EasyObject to compare images. In this example, we will check the quality of a PCB film.

You'll need first to load a reliable source image (step 1), from which two reference images (min and max) will be built (step 2). Then you'll load another image to be inspected (step 3), and perform the comparison with the min and max reference images (step 4). The differences will be detected.



High (left) and low (right) threshold reference images



In another image, differences are detected

Step 1: Load the source image

1. From the main menu, click `EasyObject`, then `Make Min Max`.
2. Click the `Open` icon of the Source Image area, and load the image file `EasyObject\FilmOk.png`.
3. Enter 'filmOk' for the name of the new image, and click `OK`.

Step 2: Build min and max reference images

1. Click `Execute`. Min and Max reference images are created, based on the source image.
   - □ `filmOkMax` is computed by dilating `filmOk` a given number of times ('geometric margin') and adding a constant ('gray level margin') to every pixel. `filmOkMin` is computed by eroding `filmOk` the same number of times and subtracting the same constant to every pixel.
   - □ The geometric margin can be seen as a position tolerance between the image to be inspected and the reference.
   - □ The gray level margin introduces a tolerance to lighting variations.

Step 3: Load an image to be inspected

1. From the main menu, click `EasyObject`, then `New EasyObject Tool`.
2. Keep the default variable name for the new object, and click `OK`.
3. In the `Encoder` tab, click the `Open` icon of the Source Image area, and load the image file `EasyObject\FilmBad.png`.
4. Enter '`filmBad`' for the name of the new image, and click `OK`

Step 4: Compare the image with the reference images

1. In the `Encoder` tab, select `ImageRange` in the segmentation method drop-down list.
2. Disable the `White Layer` check-box, and enable the `Black Layer` check-box.
3. Click the `...` button around the High Image field. Select `filmOkMax` in the drop-down list, and click `OK`.
4. Click the `...` button around the Low Image field. Select `filmOkMin` in the drop-down list, and click `OK`.

5. Click Encode. Eight differences are highlighted in the image.
6. Click Results to get further information about the detected objects. They may be further filtered and analyzed using object features selection and sorting capabilities of EasyObject.

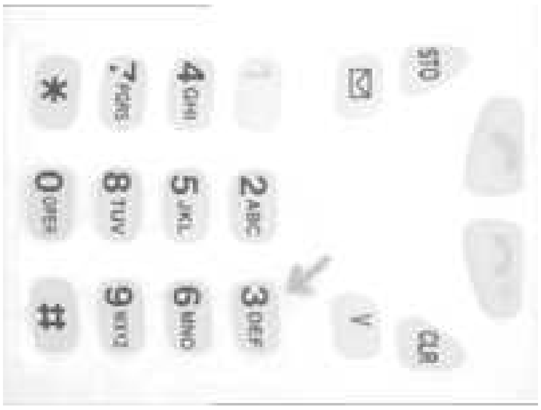
Detecting Printing Errors Using a Flexible Mask

Generating a Flexible Mask from an Encoded Image

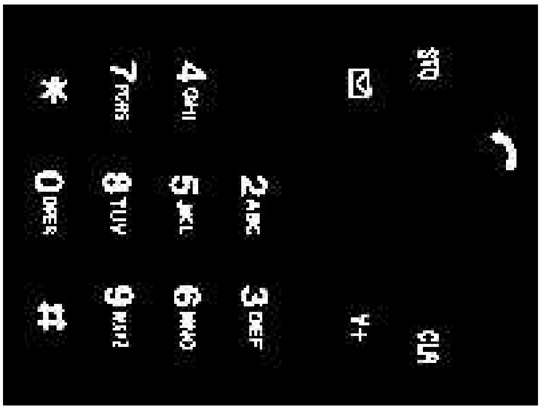
Objective

Following this tutorial, you will learn how to use a flexible mask to target and search specific areas in the image.

You'll need first to load a source image (step 1), and a flexible mask image (step 2), that can be automatically applied on the source image to separate do-care areas (that must be considered) and don't-care areas (that should not be considered). Then, you'll perform the inspection only on do-care areas (step 3).



Source image



Mask image

Results

Step 1: Load the source image

1. From the main menu, click EasyObject, then New EasyObject Tool.
2. Keep the default variable name for the new object, and click OK.
3. In the Encoder tab, click the Open icon, and load the image file EasyObject\Mobile3.jpg.
4. Keep the default variable name for the new image, and click OK.

Step 2: Load the flexible mask image

1. In the Encoder tab, click the Open icon, and load the flexible mask image file EasyObject\MobileMask.bmp.
2. Enter 'mask' for the name of the new image, and click OK. In the Mask area of the Encoder tab, notice that the mask image is selected from the drop-down list: the mask is automatically applied on the source image, because its name contains 'mask', and because it has been loaded from the coded image dialog box. The source image preview in the dialog box shows (in red diagonal lines) the don't-care area, that is the area that will be not be considered when encoding the source image.

Step 3: Inspect the image

1. In the Segmentation area of the Encoder tab, click the ... button to display the threshold dialog box.
2. Select Absolute and enter 202 for the threshold. Click OK to close the dialog box.
3. Click Encode to locate the objects (in the do-care areas only). In the source image, each object is drawn using a different color. Three printing errors can be observed:
   - The digit '7' is partially printed.
   - The '+' sign is missing.
   - The handset is printed on a lighter tone.
4. Click Results to display the statistics on each object.
   - Selecting an object in the list highlights it in the image.
   - Selecting Columns and Drawing will display more options.

2.5. EasyMatch

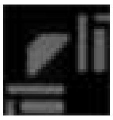
Learning a Pattern and Creating an EasyMatch Model File

Pattern Learning

Objective

Following this tutorial, you will learn how to use EasyMatch to learn a model from a reference image, and save it as an EasyMatch model file.

You'll need first to load the reference image (step 1). Then, you'll learn it as the reference model (step 2). And you'll save the model as an EasyMatch model file (step 3).



Reference image

Step 1: Load the reference image

1. From the main menu, click `EasyMatch`, then `New Match Tool`.
2. Keep the default variable name for the new matcher object, and click `OK`.
3. In the `Learning` tab, click the `Open` icon, and load the image file `EasyMatch\FrameModel.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Learn the reference image

- In the `Learning` tab, click `Learn` to acquire the model pattern.

Step 3: Save the model file

1. In the `Learning` tab, click the `Save As...` button.
2. Type a file name for the new EasyMatch model file. Its extension will be `.mch`.
3. Click `Save`.

Matching a Pattern According to a Model File

Pattern Matching and Retrieving Results

Objective

Following this tutorial, you will learn how to use EasyMatch to load an EasyMatch model file, and search for occurrences of the pattern in an image.

You'll need first to load the model file, and a source image where the model will be searched for (steps 1-2). Then the pattern matching is fully automatic (step 3).

Occurrences of the model are automatically highlighted

Step 1: Load the model file

1. From the main menu, click `EasyMatch`, then `New Match Tool`.
2. Keep the default variable name for the new matcher object, and click `OK`.
3. In the `Learning` tab, click `Load...` to open the model file `EasyMatch\Switch.MCH`. The model contains all necessary information about the pattern we are searching for.

Step 2: Load a source image

1. In the `Matching` tab, click the `Open` icon, and load the image file `EasyMatch\Switch1.tif`.
2. Keep the default variable name for the new image object, and click `OK`.

Step 3: Perform the pattern matching

1. The pattern matching is automatically performed on the source image, and the matching occurrences are highlighted. Clicking `Execute` will insert the corresponding code into the script windows.
2. Further information about each occurrence can be found by clicking `Results`.
3. Click in a row to see the corresponding occurrence highlighted in the image.

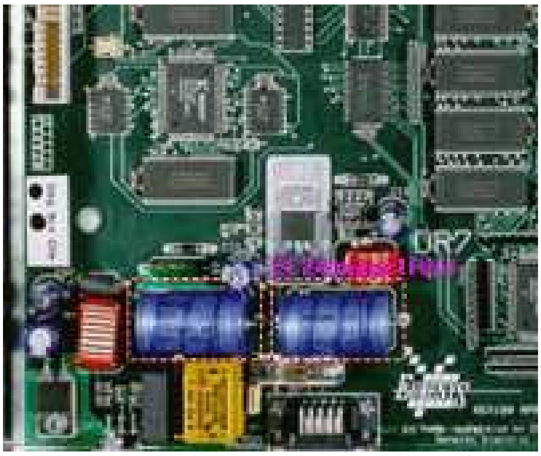
Learning a Pattern According to an ROI

Pattern Learning

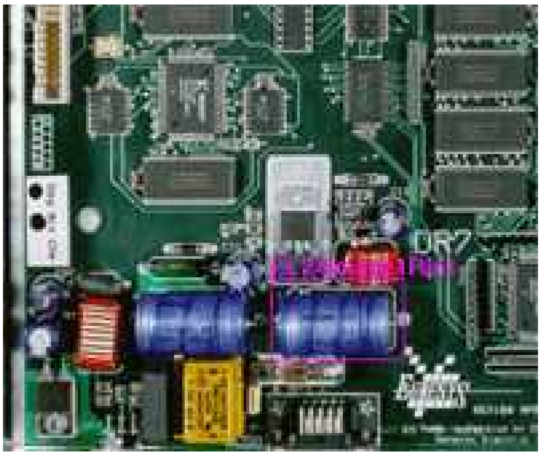
Objective

Following this tutorial, you will learn how to use EasyMatch to learn a model from an ROI in a source image, and to perform pattern matching on the same image.

You'll need first to load the source image, and define an ROI inside (steps 1-2). Then, you'll have to learn the model, using this ROI (step 3). Finally, you'll perform pattern matching in the source image (step 4), and will find additional occurrences of the model.



ROI that will be learned



Occurrences matching the model ROI

Step 1: Load the source image

1. From the main menu, click `Image`, then `Open`.
2. Load the image file `EasyMatch\BOARD.JPG`.
3. Keep the default variable name for the new image object, and click `OK`.

Step 2: Define an ROI

1. Right-click in the image, and select `New ROI...` from the contextual menu.

2. Keep the default variable name for the new ROI object, and click `OK`. A default ROI is placed over the image (blue rectangle with handles).

The ROI management dialog box is opened.

3. Resize the ROI and move it around one of the blue capacitors at the lower left part of the image.

Step 3: Learn a model from the ROI

1. From the main menu, click `EasyMatch`, then `New Match Tool`.
2. Keep the default variable name for the new matcher object, and click `OK`.
3. In the `Learning` tab of the matcher dialog box, select the ROI object from the Source Image drop-down list, and click `Learn` to acquire the model pattern.

Step 4: Match the pattern

1. In the `Matching` tab, increase the `Max Occurrences` field to 2.
2. Select the image object from the Source Image drop-down list.
3. Click `Execute`. The occurrences of the learned model are highlighted in the source image.
4. Further information about each occurrence can be found by clicking `Results`.
5. Click in a row to see the corresponding occurrence highlighted in the image.

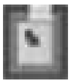
Improving the Score of Matching Instances by Using "Don't Care Areas"

Setting Search Parameters

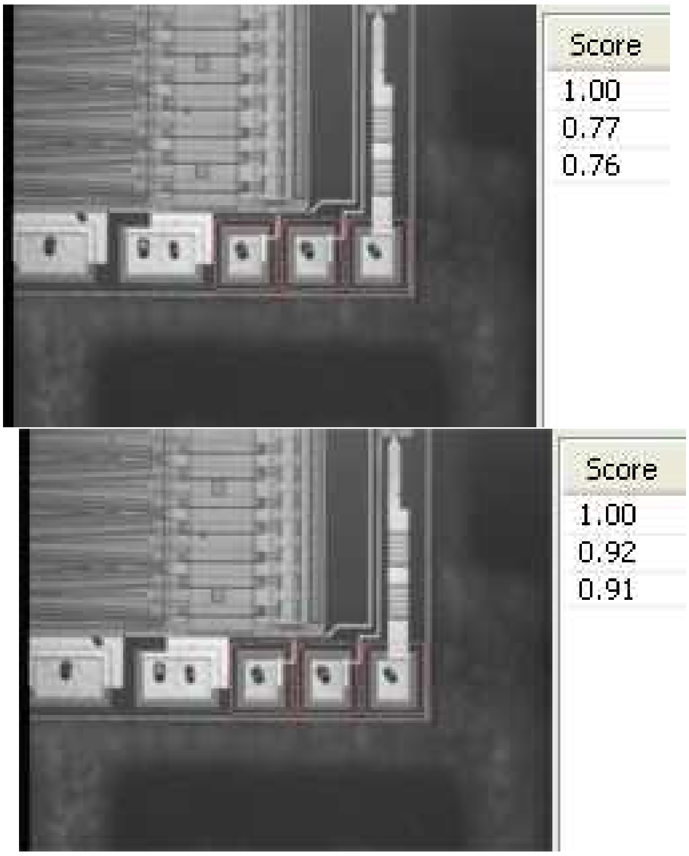
Objective

Following this tutorial, you will learn how to use EasyMatch to handle "don't care areas". "Don't care areas" help to define in the image the meaningful features only, by masking the areas that might change from image to image.

You'll need first to load a reference image and learn the reference model (steps 1-2). Then you'll perform automatic pattern matching of instances of the reference model, without using "don't care areas" (step 3). As the matching scores of the found instances are not high enough, you'll define a "don't care area" on the reference model, and restart the detection. The matching scores are much better (steps 4-5).



Reference model



Score
1.00
0.77
0.76

Score
1.00
0.92
0.91

Found instances and matching scores, without (left) and with (right) using "don't care areas"

Step 1: Load the reference image

1. From the main menu, click `EasyMatch`, then `New Match Tool`.
2. Keep the default variable name for the new Matcher object, and click `OK`. The Matcher management dialog box is opened.
3. In the `Learning` tab, click the `Open` icon, and load the image file `EasyMatch\Die Pad Model 1.bmp`.
4. Keep the default variable name for the new Image object, and click `OK`.

Step 2: Learn the reference model

- Click `Learn` to acquire the model pattern.

Step 3: Detect instances of the reference model without "don't care areas"

1. In the `Matching` tab, click the `Open` icon, and load the image file `EasyMatch\Die Pad 1.bmp`.
2. Keep the default variable name for the new Image object, and click `OK`. An instance matching the reference model is highlighted.
3. Increase the `Max Occurrences` field to 3. Enable the `Sub-Pixel Interpolate` check-box to increase the matching precision.

4. Enter '-10.0' as the Minimum Angle (Deg). (Check that the angle is displayed in degrees. If not, select the angles unit from `View` > `Option` menu.)
5. Enter '10.0' as the Maximum Angle (Deg).
6. Click `Execute`. The pattern locations are highlighted in the source image.
7. Click `Results` to see the matching score of each found instance. The last two scores are rather bad. This is mainly due to the bright rectangle on the upper part of the reference image we have learned. We can improve the scores by using a "don't care area" to mask this bright area.

Step 4: Define the "don't care area"

1. In the `Don't Care Areas` tab, select the `Rectangle` radio button from the `Blacken Inside` group.
2. In the reference image, move your mouse pointer on the lower left corner of the bright rectangle, left-click and drag the don't care area (rectangle with red stripes) to the upper right corner of the bright rectangle to mask out this area.
3. In the `Don't Care Areas` tab, click `Learn`.

Step 5: Detect instances of the reference model with "don't care areas"

1. In the `Matching` tab, click `Execute`. The instances matching the reference model are still highlighted.
2. Click `Results` to compare the new matching scores. They are much better.

2.6. EasyFind

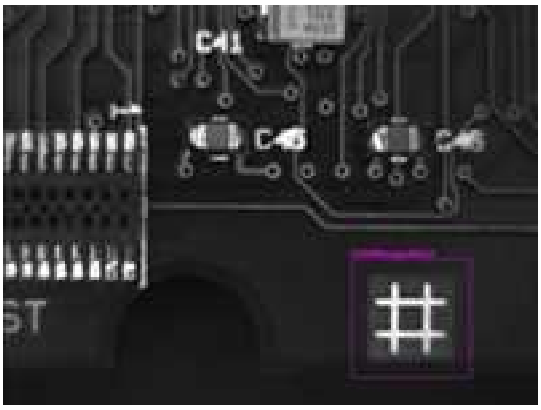
Detecting Highly-Degraded Occurrences of a Reference Model in Multiple Files

Pattern Finding and Retrieving Results

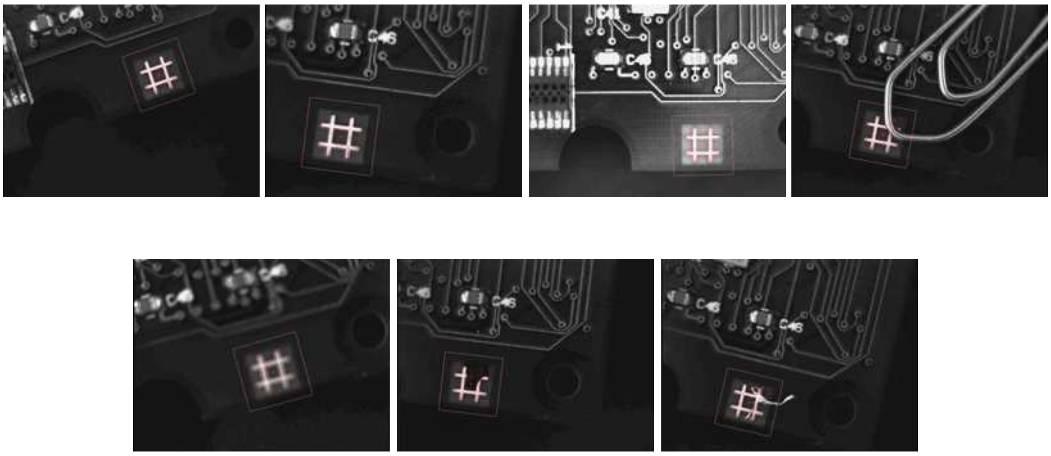
Objective

Following this tutorial, you will learn how to use EasyFind to detect in multiple images highly-degraded occurrences of a reference model. The degradation can be due to noise, blur, occlusion, missing parts or unstable illumination conditions.

You'll need first to load a reference image, define an ROI where EasyFind will learn the reference model, and set rotation and scaling tolerances for the expected occurrences to search for (steps 1-4). Then you're ready to open multiple files, and perform automatic detection of occurrences (even highly-degraded) of the reference model (steps 5-6).



Reference model



Occurrences of the reference model are found, even if highly-degraded

Step 1: Load the reference image

1. From the main menu, click `EasyFind`, then `New EasyFind Tool`.
2. Keep the default variable name for the new PatternFinder object, and click `OK`. The PatternFinder management dialog box is opened.
3. In the `Model` tab, click the `Open` icon, and load the image file `EasyFind\Fiducial 1.tif`.
4. Keep the default variable name for the new image object, and click OK.

Step 2: Create an ROI to define the reference model on the reference image

1. In the image, right-click and select `New ROI...` item from the menu.
2. Keep the default variable name for the new ROI object, and click `OK`. A default ROI is placed over the image (blue rectangle with handles). The ROI management dialog box is opened.
3. Drag the ROI over the reference model and resize it using its handles. Alternatively, enter the following coordinates in the ROI dialog box: 500, 365, 170, 170 for OrgX, OrgY, Width, and Height respectively, and click `Close`.

Step 3: Learn the reference model

1. In the PatternFinder `Model` tab, select the ROI object from the source image drop-down list, and click `Learn`. The reference model is perfectly detected (green edges).
2. In the PatternFinder `Search Field` tab, select the Image object from the source image drop-down list. Tick the Draw Features check-box.

The model location and feature points are highlighted in the source image.

Step 4: Set rotation and scaling tolerances

- In the PatternFinder `Allowances` tab, set both angle tolerance and scale tolerance to 25.0.

Step 5: Select multiple images

1. In the PatternFinder `Search Field` tab, click the `Open` icon. Select the images files `EasyFind\Fiducial 2.tif` to `Fiducial 8.tif` (use the shift key to select multiple files), and click `Open`.
2. Keep the default variable name for the new Image object, and click `OK`. The last image appears. The reference model is found, even if highly-degraded.
3. Detection of the reference model is automatically performed. It is not necessary to click Find once a new image appears, as inspection is automatically performed. However, clicking `Find` will insert the corresponding code into the script windows.
4. Click `Results` to find more information about the found instance.

Step 6: Browse multiple images

- In the PatternFinder `Search Field` tab, click the `Load Next` or `Load Previous` icons.

The image files appear, and the reference model is automatically detected.

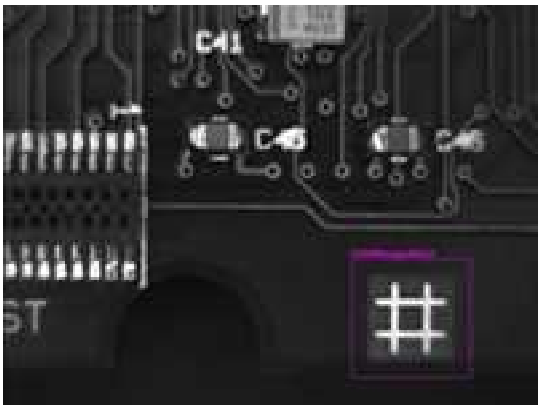
Improving the Score of Found Instances by Using "Don't Care Areas"

Setting Search Parameters

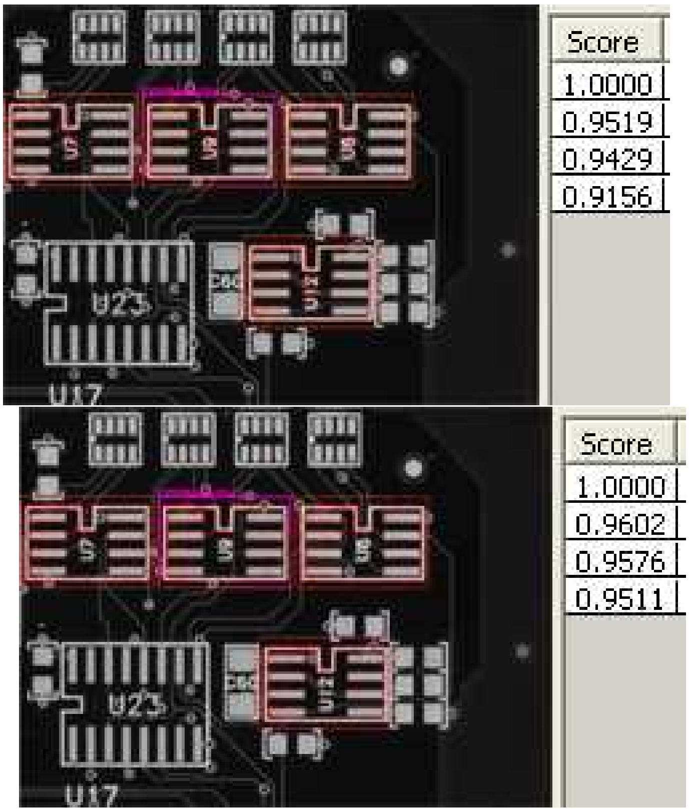
Objective

Following this tutorial, you will learn how to use EasyFind to handle "don't care areas" in geometric pattern matching. "Don't care areas" help to define in the image the meaningful features only, by masking the areas that might change from image to image, such as text and numbers.

You'll need first to load a reference image, define an ROI where EasyFind will learn the reference model, and set a rotation tolerance for the expected instances to search for (steps 1-4). Then you're ready to perform automatic detection of instances of the reference model, without using "don't care areas" (step 5). As the matching scores of the found instances are not high enough, you'll define a "don't care area" on the reference model, and restart the detection. The matching scores are slightly better (steps 6-7).



Reference model



Score
1.0000
0.9519
0.9429
0.9156

Score
1.0000
0.9602
0.9576
0.9511

Found instances and matching scores, without (left) and with (right) using "don't care areas"

Step 1: Loading the reference image

1. From the main menu, click EasyFind, then New EasyFind Tool.
2. Keep the default variable name for the new PatternFinder object, and click OK. The PatternFinder management dialog box is opened.
3. In the Model tab, click the Open icon, and load the image file EasyFind\Solder Pad 1.tif.
4. Keep the default variable name for the new Image object, and click OK.

Step 2: Creating an ROI to define the reference model on the reference image

1. In the image, right-click and select `New ROI...` item from the menu.
2. Keep the default variable name for the new ROI object, and click `OK`. A default ROI is placed over the image (blue rectangle with handles). The ROI management dialog box is opened.
3. Drag the ROI over the reference model and resize it using its handles. Alternatively, enter the following coordinates in the ROI dialog box: 200, 130, 190, 130 for OrgX, OrgY, Width, and Height respectively, and click `Close`.

Step 3: Learning the reference model

1. In the PatternFinder `Model` tab, select the ROI object from the source image drop-down list, and click `Learn`. The reference model is detected.
2. In the PatternFinder `Search Field` tab, select the Image object from the source image drop-down list. Tick the `Draw Features` check-box. The model location and feature points are highlighted in the source image.

Step 4: Setting a rotation tolerance

1. In the PatternFinder `Allowances` tab, set the angle tolerance to 5.0.

Step 5: Detecting instances of the reference model without "don't care areas"

1. In the PatternFinder `Search Field` tab, set `Max Instances` to 4, and click `Find`. The instances matching the reference model are highlighted.
2. Click `Results` to see the matching score of each found instance. Even though the scores are good, we can still improve them slightly by using a "don't care area" to mask the text appearing in the learned pattern.

Step 6: Defining the "don't care area"

1. In the PatternFinder `Don't Care Areas` tab, select the `Rectangle` radio button from the `Blacken Inside` group.
2. In the ROI defining the reference model, move your mouse pointer on the top-left corner of the text "U9", left-click and drag the don't care area (rectangle with red stripes) to mask out the text.

Step 7: Detecting instances of the reference model with "don't care areas"

1. In the PatternFinder `Don't Care Areas` tab, click `Learn`, and then click `Find`.

The instances matching the reference model are still highlighted, but the text is not found anymore.

2. Click `Results` to compare the new matching scores.

They are slightly better.

2.7. EasyGauge

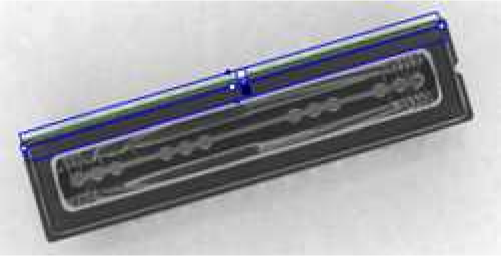
Measuring the Rotation Angle of an Object

Line Fitting

Objective

Following this tutorial, you will learn how to use EasyGauge to measure the rotation angle of a CCD sensor package. As we only need to retrieve an angle value, it's not required to work in a calibrated field of view. All geometrical parameters and results will be express as numbers of pixels.

You'll need to load the source image (step 1), and attach a line fitting gauge (step 2). The inspection is automatically performed (step 3).



Line fitting gauge

Step 1: Load the source image

1. From the main menu, click `EasyGauge`, then `New World Shape`.
2. Keep the default variable name, and click `OK`.
3. From the `Gauges` tab, click the `Open` icon, and load the image file `EasyImage\CCD.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Attach a line gauge to the image

1. In the `Gauges` tab of the world shape dialog box, right-click the world shape icon, select `New` > `Line Gauge` from the contextual menu.
2. Keep the default variable name, and click `OK`. The line location gauge appears on the image. It consists of the following elements:
   - □ A blue line segment along which the transitions search is carried out.
   - □ Five white handles, allowing the user to move and rotate the segment.
   - □ A gray arrow, indicating in which direction the segment is traversed.
   - □ Black and white rectangles, indicating which kind of transition is searched for.
   - □ Green line, indicating the transition points found (if any).
3. Using the handles, move, rotate and extend the line gauge, so that it is positioned on the upper edge of the CCD package, with the gray arrow pointing downwards.

4. In the Measurement tab of the line gauge dialog box, choose 'White to Black' from the Type dropdown list and 'From Begin' from the Choice dropdown list.

Step 3: Perform the inspection

1. The image is automatically inspected. However, clicking Process in the world shape dialog box will insert the corresponding code into the script window.
2. Click the Results tab to retrieve the measured angle value.
3. To see the individual fitting points, select the Points checkbox under the Draw Samples area.

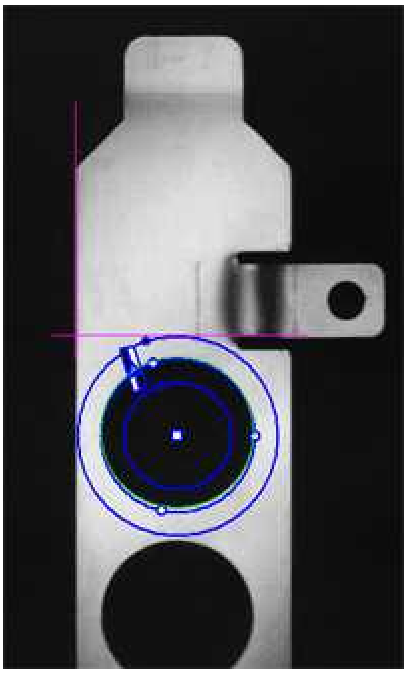
Measuring the Diameter of a Circle

Circle Fitting

Objective

Following this tutorial, you will learn how to use EasyGauge to measure the diameter of a circle in an image.

You'll first load an image for calibration —a dot grid— (step 1), and calibrate the field of view (step 2). Then you'll load the source image for inspection (step 3), and attach a circle fitting gauge (step 4). The inspection is automatically performed (step 5). All measurement results are expressed in physical units..



Measuring the diameter of a circle

Step 1: Load the calibration image

1. From the main menu, click EasyGauge, then New World Shape.
2. Keep the default variable name, and click OK.

3. In the `Dot Grid Calibration` tab, click the `Open` icon, and load the image file `EasyGauge\Dot Grid 1.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Calibrate the field of view

- Click `Calibrate`. From now on, the field of view is calibrated, and all results will be expressed in physical units.

Step 3: Load the source image

1. From the `Gauges` tab, click the Open icon, and load the image file `EasyGauge\Bracket 6.tif`.
2. Keep the default variable name, and click `OK`.

Step 4: Attach a circle gauge to the image

1. In the `Gauges` tab of the world shape dialog box, right-click the frame shape icon, select `New` > `Circle Gauge` from the contextual menu.
2. Keep the default variable name, and click `OK`.
3. The circle location gauge appears on the image. It consists of the following elements:
   - A blue ring in which the circle is searched for.
   - A blue nominal circle.
   - Six white handles, allowing the user to move and rotate the segment.
   - A gray arrow, indicating in which direction the segment is traversed.
   - Black and white rectangles, indicating which kind of transition are searched for.
   - A green arc of circle, indicating the circle found (if any).
4. Using the handles, drag the circle fitting gauge around the upper bracket hole. Adjust the nominal circle on the hole edge and extend the searching area if necessary.
5. In the `Measurement` tab of the circle gauge dialog box, select 'From Begin' from the Choice dropdown list.

Step 5: Perform the inspection

1. The image is automatically inspected. However, clicking `Process` in the world shape dialog box will insert the corresponding code into the script window.
2. Click the `Results` tab to retrieve the measured diameter. All measurement results are expressed in physical units.

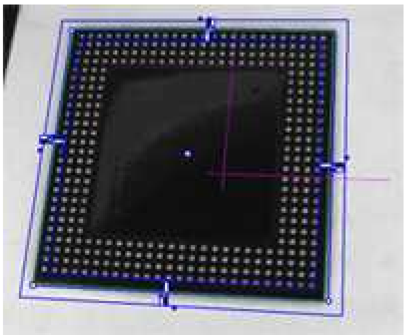
Measuring a Distorted Rectangle

Rectangle Fitting

Objective

Following this tutorial, you will learn how to use EasyGauge to perform measurements on a distorted rectangle component.

To obtain measurement results in physical units, we need to work in a calibrated field of view. You'll need first to load an image for calibration —a dot grid— (step 1), and calibrate the field of view (step 2). Then you'll load the distorted image (step 3), and attach a rectangle fitting gauge (step 4). The inspection is automatically performed (step 5). All measurement results are expressed in physical units.



Measuring a distorted rectangle

Step 1: Load the calibration image

1. From the main menu, click `EasyGauge`, then `New World Shape`.
2. Keep the default variable name, and click `OK`.
3. In the `Dot Grid Calibration` tab, click the `Open` icon, and load the image file `EasyGauge\Dot Grid 5.tif`. This dot grid has been acquired in the same conditions and has the same distortion as the image we want to inspect.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Calibrate the field of view

- Click `Calibrate`. From now on, the field of view is calibrated, and all results will be expressed in physical units.

Step 3: Load the distorted image

1. From the `Gauges` tab, click the `Open` icon, and load the image file `EasyGauge\Distorted Component.tif`.
2. Keep the default variable name, and click `OK`.

Step 4: Attach a rectangle gauge to the image

1. In the `Gauges` tab, right-click the world shape icon, and select `New` > `Rectangle Gauge` from the contextual menu.
2. Keep the default variable name, and click `OK`. The rectangle gauge dialog box is opened, and the rectangle gauge is drawn on the image. It consists of the following elements:
   - A blue rectangular ring in which the rectangle is searched for.
   - A blue nominal rectangle.
   - Eleven white handles, allowing the user to move and extend the search area.
   - Gray arrows, indicating in which direction segments are examined.
   - Black and white rectangles, indicating which kind of transition are searched for.

- □ A green rectangle, indicating the rectangle found (if any).

3. Due to the perspective effect, the rectangle gauge doesn't look like a rectangle. Using the central handle, move the rectangle gauge in the image and observe the rectangle deformation. Due to the calibration, the rectangle gauge shape adapts to the field of view deformation. Extend the searching area, and adjust the nominal rectangle on the component edges.
4. In the `Measurement` tab of the rectangle gauge dialog box, select 'White To Black' from the Type dropdown list and 'From Begin' from the Choice dropdown list.

Step 5: Perform the inspection

1. The image is automatically inspected. However, clicking `Process` in the world shape dialog box will insert the corresponding code into the script window.
2. Click the `Results` tab to retrieve the measured X and Y sizes. All measurement results are expressed in physical units.

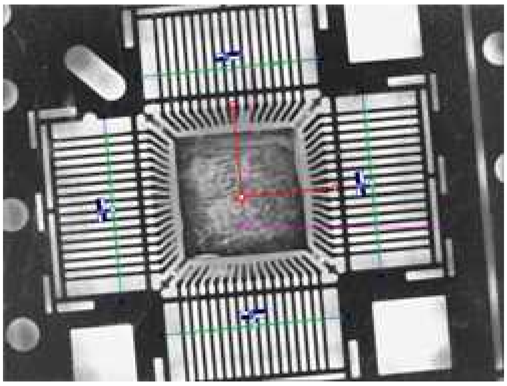
Locating Points Regarding to a Coordinate System

Point Location

Objective

Following this tutorial, you will learn how to use EasyGauge to perform lead frames inspection. This operation determines the dimension, position, curvature, size, angle or diameter of the lead frames with an excellent accuracy. Robustness is ensured by powerful edge-point selection mechanisms that are intuitive and easy to tune, allowing measurement in cluttered images.

You'll first load an image for calibration —a dot grid— (step 1), and calibrate the field of view (step 2). Then you'll load the lead frame image (step 3), and set a coordinate system (a frame shape). Regarding to this coordinate system, you can define point gauges (steps 5-6). Finally, you'll load another lead frame image, that has a slight angle deviation, so the coordinate system has to be rotated (steps 7-8). The inspection is then automatically performed (step 9). All measurement results are expressed in physical units.



Four point gauges over four sets of leads

Step 1: Load the calibration image

1. From the main menu, click `EasyGauge`, then `New World Shape`.

2. Keep the default variable name, and click OK.
3. In the Dot Grid Calibration tab, click the Open icon, and load the image file `EasyGauge\Dot Grid 2.tif`.
4. Keep the default variable name for the new image object, and click OK.

Step 2: Calibrating the field of view

1. Click Calibrate. From now on, the field of view is calibrated, and all results will be expressed in physical units.

Step 3: Loading a lead frame image

1. From the Gauges tab, click the Open icon, and load the image file `EasyGauge\Lead Frame 1.tif`.
2. Keep the default variable name, and click OK.

Step 4: Setting a coordinate system

1. In the Gauges tab of the world shape dialog box, right-click the world shape icon, select New > Frame Shape from the contextual menu.
2. Keep the default variable name, and click OK. The frame shape icon appears in the world shape dialog box.
3. Drag the frame shape center approximately to the square center of the image.

Step 5: Attaching a point gauge to the frame shape

1. In the Gauges tab of the world shape dialog box, right-click the frame shape icon, select New > Point Gauge from the contextual menu.
2. Keep the default variable name, and click OK. The point location gauge appears on the image. It consists of the following elements:
   - A blue line segment along which the transitions search is carried out.
   - Three white handles, allowing the user to move and rotate the segment.
   - A gray arrow, indicating in which direction the segment is traversed.
   - Black and white rectangles, indicating which kind of transition is searched for.
   - Green crosses, indicating the transition points found (if any).
3. Place the point location gauge over a set of leads: in the Position tab of the point gauge dialog box, set the Center Y property to 7, and the Tolerance property to 5.

Step 6: Attaching other point gauges to the frame shape

1. From the Gauges tab of the world shape dialog box, create three other point gauges (refer to step 5).
2. Place these point location gauges over the remaining sets of leads :
   - Center Y = -7, Tolerance = 5
   - Center X = 7, Tolerance = 5, Angle = 90
   - Center X = -7, Tolerance = 5, Angle = 90

Step 7: Loading another lead frame image

1. From the Gauges tab, click the Open icon, and load the image file `EasyGauge\Lead Frame 2.tif`.
2. Keep the default variable name for the new image, and click OK.

Step 8: Tuning the coordinate system

1. In the frame shape dialog box, set the Angle property to 5.8.
2. Drag the frame shape center approximately to the square center of the image. All point location gauges automatically follow.

Step 9: Performing the inspection

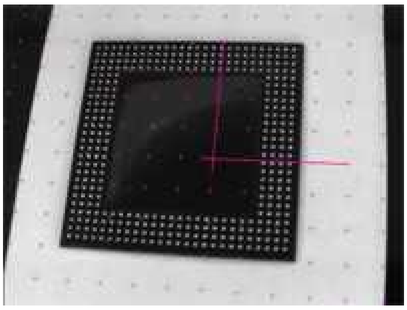
1. The image is automatically inspected. However, clicking Process in the world shape dialog box will insert the corresponding code into the script window.
2. From the point gauge dialog boxes, click the Results tab to retrieve the located points coordinates. All measurement results are expressed in physical units. The values refer to the frame shape system.

Unwarping a Distorted Image

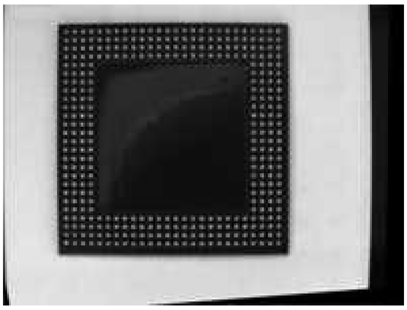
Objective

Following this tutorial, you will learn how to use EasyGauge to perform grid calibration, and unwarp a distorted image.

You'll first load an image for calibration —a dot grid— (step 1), and calibrate the field of view (step 2). Then you'll load the distorted image (step 3), and perform the unwarping operation (step 4).



Distorted image



Unwarped image

Step 1: Load the calibration image

1. From the main menu, click `EasyGauge`, then `New World Shape`.
2. Keep the default variable name, and click `OK`.
3. In the `Dot Grid Calibration` tab, click the `Open` icon, and load the image file `EasyGauge\Dot Grid 5.tif`. This dot grid has been acquired in the same conditions and has the same distortion as the image we want to unwarp.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Calibrate the field of view

- Click `Calibrate`.

Step 3: Load the distorted image

1. From the `Unwarping` tab, click the `Open` icon, and load the image file `EasyGauge\Distorted component.tif`.
2. Keep the default variable name, and click `OK`.

Step 4: Unwarp the distorted image

1. In the Destination Image area, click `New` icon to create a new image.
2. Keep the default image settings, and click `OK`.
3. Select `Interpolate` checkbox to improve resulting image quality.
4. Click `Unwarp`. In the destination image, all distortions are corrected.

2.8. EasyOCR

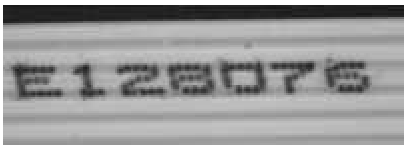
Learning Characters and Creating an EasyOCR Font

Learning Characters

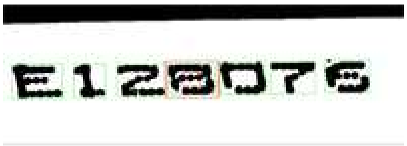
Objective

Following this tutorial, you will learn how to use EasyOCR to learn new characters and save them in an EasyOCR font.

You'll need first to load a source image (step 1). Then you'll set the segmentation parameters to isolate each character (step 2). Each character will have to be learnt (step 3), and finally you'll save all the learnt characters as a font file (step 4). You can also add new characters to an existing font if needed (step 5).



Source image



The image is segmented so that all the characters are detected

All the characters have been learn

Step 1: Load the source image

1. From the main menu, click `EasyOCR`, then `New OCR Tool`.
2. Keep the default variable name for the new OCR object, and click `OK`.
3. In the `Source Image` tab, click the `Open` icon of the Source Image area, and load the image file `EasyOCR\FlatCable\FlatCable1.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Set segmentation parameters

1. Select the `Segmentation Parameters` tab, and move the red frame in the image above a character.
2. Tune each property to get a green bounding box around each character:
   - □ `threshold value` = 113
   - □ `characters` color = Black on White
   - □ `min width` = 36
   - □ `min height` = 31

- □ spacing = 4
- □ max width = 98
- □ max height = 72
- □ noise area = 9

Step 3: Learn new characters

1. Select the Learn tab, and click the character E in the image. You are then prompted to identify the character along with its class. Enter E in the character field, and select the 'EOcrClass_Uppercase' class. Click OK. Whenever a character has been added to the current font, its bounding box turns yellow.
2. Click the character 1 in the image. Enter 1 in the character field, and select the 'EOcrClass_Digit' class. Click OK.
3. Proceed with remaining characters.

Step 4: Save the EasyOCR font

- In the Font File tab, click the Save As... button. Type a file name for the new EasyOCR font file. Its extension will be .ocr. Finally, click Save.

Step 5: Add characters to an existing font

1. In the Source Image tab, click the Open icon of the Source Image area, and load the image file `EasyOCR\FlatCable\FlatCable2.tif`.
2. Keep the default variable name for the new image object, and click OK.
3. In the Recognition tab, click Execute. Characters 2 and 8 are read correctly, but A, W and G are not (low confidence score). They don't belong to the font.
4. Select the Learn tab, and learn the characters A, W, and G (refer to step 3).
5. Then save the font again (refer to step 4). The new characters have been added.

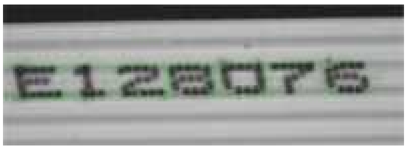
Recognizing Characters According to a Font

"Recognizing Characters According to a Font" on page 59

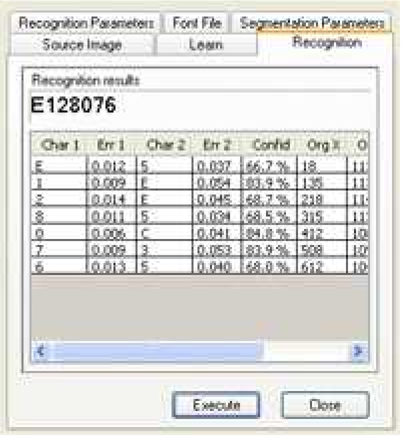
Objective

Following this tutorial, you will learn how to use EasyOCR to recognize characters, regarding to a specific font.

You'll need first to load a source image (step 1), and an EasyOCR font file (step 2). Then you'll perform the characters recognition (step 3).



Characters matching the font are automatically detected



Recognition results

E128076

Char 1	Err 1	Char 2	Err 2	Confid	Org X	O
E	0.012	5	0.037	66.7 %	18	11
1	0.009	E	0.054	83.9 %	135	11
2	0.014	E	0.045	68.7 %	218	11
8	0.011	5	0.034	68.5 %	315	11
0	0.006	C	0.041	84.8 %	412	10
7	0.009	3	0.053	83.9 %	508	10
6	0.013	5	0.040	68.0 %	612	10

Results after explicit recognition

Step 1: Load the source image

1. From the main menu, click `EasyOCR`, then `New OCR Tool`.
2. Keep the default variable name for the new OCR object, and click `OK`.
3. In the Source Image tab, click the `Open` icon of the Source Image area, and load the image file `EasyOCR\FlatCable\FlatCable1.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Load the font file

- In the `Font File` tab, click `Load`, and select the font file `EasyOCR\FlatCable\FlatCable.ocr`. In the image, the detected characters are highlighted in green.

Step 3: Recognize the characters

- In the `Recognition` tab, click `Execute` to trigger the recognition of the detected characters. The recognized characters appear in the Recognition results area. Further information about each character can be found in the table.

2.9. EasyBarCode

Reading Bar Codes Automatically

Reading a Bar Code

Objective

Following this tutorial, you will learn how to perform automatic reading of multiple bar codes.

You'll need first to load multiple source images (step 1). The reading is then automatically performed on each image (step 2).

Each bar code is automatically detected and decoded

Step 1: Load the source images

1. From the main menu, click `EasyBarCode`, then `New BarCode Tool`.
2. Keep the default variable name, and click `OK`.
3. In the `AutoRead` tab, click the `Open` icon of the Source Image area, and load the image files `EasyBarCode\Barcode 01.tif` to `Barcode 10.tif`. Use the shift key to select multiple files.
4. Keep the default variable name, and click `OK`. The last image appears.

Step 2: Read the bar codes automatically

1. The bar code is automatically detected and decoded. The graphic result appears on the image, while the data content and the corresponding symbology are displayed in the Decoded Symbology area. It is not necessary to click `Read` once a new image appears. However, clicking `Read` will insert the corresponding code into the script window.
2. In the `Results` tab, find more information about the bar code. As a bar code might have a meaning under different symbologies, all possible contents are listed by decreasing likeliness.
3. In the `AutoRead` tab, click the `Load Next` and `Load Previous` icons to browse through images 01 to 06.

The image files appear, and each bar code is automatically detected and decoded. The bar code properties are updated.

4. To decode the remaining bar codes, we have to enable the additional symbologies.

5. In the `Symbologies` tab, click the `Toggle All` button of the Additional area.
6. In the `AutoRead` tab, click the `Load Next` and `Load Previous` icons and browse the remaining images.

Note about EasyBarCode2

Since the version 22.04, **Open eVision Studio** showcases **EasyBarCode2**, our new bar code reader: it is faster, reads more complex bar codes, can read several bar codes on an image and supports more symbologies.

Don’t hesitate to check it out!

- To read Barcode 02 (MSI), Barcode 05 (Code 25 Interleaved), Barcode 07 (Code 25 Datalogic), Barcode 08 (Code 11), Barcode 10 (Code CIP) and BarCode 12 (PharmaCode), enable the corresponding symbology in the symbologies tab.
- To read Barcode 03 (Code 39), disable the bar code validation in the `AutoRead` tab.

2.10. EasyMatrixCode

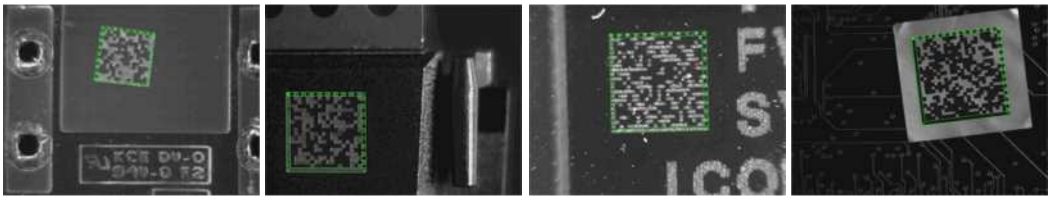
Reading Data Matrix Codes Automatically

Automatic Reading

Objective

Following this tutorial, you will learn how to use EasyMatrixCode to detect and decode automatically Data Matrix codes in multiple files.

You'll need first to load multiple source images (step 1). The reading is then automatically performed on each image (step 2). You can also grade the printing quality of each matrix code (step 3).



Each Data Matrix code is automatically detected and decoded

Step 1: Load the source images

1. From the main menu, click `EasyMatrixCode`, then `New MatrixCode Tool`.
2. Keep the default variable name for the new matrix code reader, and click `OK`.

3. In the `Read` tab, click the `Open` icon of the Source Image area, and load the image files `EasyMatrixCode\AutoRead\AutoRead 01.tif` to `AutoRead 04.tif`. Use the shift key to select multiple files.
4. Keep the default variable name for the new image, and click OK. The last image appears.

Step 2: Read the Data Matrix codes automatically

1. The Data Matrix code is automatically detected and decoded. The matrix code reference corner is highlighted with a bold cross mark. It is not necessary to click `Read` once a new image appears. However, clicking `Read` will insert the corresponding code into the script window.
2. In the `Results` area, find more information about the matrix code, such as the decoded string.
3. In the `Read` tab, click the `Load Next` and `Load Previous` icons. The image files appear, and each Data Matrix code is automatically detected and decoded. The matrix code properties are updated. If no Data Matrix code could been located in the image, an error message is displayed in the Message field.

Step 3: Grade Data Matrix code printing quality

1. In the `Print Quality` tab, select the `Compute Grading` check-box.
2. Click `Apply`.

For each printing quality parameter, the corresponding value and its grade equivalent appear. An A grade means a good quality, while an F grade indicates a poor one.

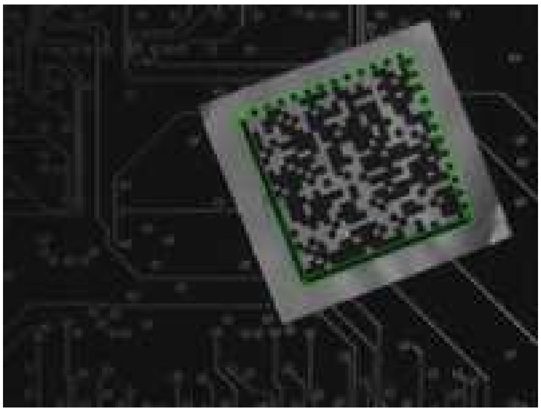
Learning a Data Matrix Code and Creating an EasyMatrixCode Model File

Reading with Prior Learning

Objective

Following this tutorial, you will learn how to use EasyMatrixCode to learn a Data Matrix code, and save it as an EasyMatrixCode model file.

You'll need first to load a source image (step 1), and learn the matrix code (step 2). Then you'll save the learned matrix code as a new model file (step 3). You can also add new learned matrix codes to an existing model if needed (step 4).



The Data Matrix code has been learned

Step 1: Load the source image

1. From the main menu, click `EasyMatrixCode`, then `New MatrixCode Tool`.
2. Keep the default variable name for the new matrix code reader object, and click `OK`.
3. In the `Learn` tab, click the `Open` icon of the Source Image area, and load the image file `EasyMatrixCode\Label\Label 4.tif`.
4. Keep the default variable name for the new image object, and click `OK`.

Step 2: Learn the Data Matrix code

- In the `Learn` tab, click `Learn`.

The Data Matrix code is detected and decoded without error.

The graphical result appears on the image.

The properties of the learned matrix code are updated in the dialog box.

Step 3: Save the model file

1. In the `Learn` tab, click the `Save As...` button.
2. Type a file name for the new EasyMatrixCode model file. Its extension will be `.mx2`.
3. Click `Save`.

Step 4: Learning more Data Matrix codes

1. In the `Read` tab, click the `Open` icon of the source image area, and load the image file `EasyMatrixCode\PCB Code\PCB Code 3.jpg`.
2. Keep the default variable name for the new image object, and click `OK`.
3. An error message is displayed in the Message area of the `Read` tab. The matrix code can not be read, since the reader uses the model from the "Label 4" image. You need to learn the "PCB Code 3" matrix code, and add it to the model.
4. In the `Learn` tab, click `Learn More`. The Data Matrix code is detected and decoded without error. The graphical result appears on the image. The properties of the learned matrix code are updated in the dialog box.

5. Using `Learn More` rather than Learn involves that the "Label 4" model is not replaced by the "PCB Code 3" model, but both are now included in the same model. In the Read tab, the "PCB Code 3" matrix code is correctly read. Select the "Label 4" image in the drop-down list of the source image area. The "Label 4" matrix code is still read without error, which means that both learned matrix codes have been kept.
6. Finally, save the model again (refer to step 3). The new matrix code has been added.