Open eVision
Release 2.13.0
Terms of Use

EURESYS s.a. shall retain all property rights, title and interest of the documentation of the hardware and the software, and of the trademarks of EURESYS s.a.

All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners.

The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, brands or documentation of EURESYS s.a. contained in this book, is not allowed without prior notice.

EURESYS s.a. may modify the product specification or change the information given in this documentation at any time, at its discretion, and without prior notice.

EURESYS s.a. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS s.a. or resulting of omissions or errors in this documentation.

This documentation is provided with Open eVision 2.13.0 (doc build 1136).

www.euresys.com
Contents

1. Release Summary .......................................................... 4
2. Release Specifications ..................................................... 8
3. Release Details .............................................................. 10
   3.1. Updated Development Tools Support ................................ 10
   3.2. New and Improved Features ........................................ 10
   3.3. Breaking Changes ..................................................... 20
   3.4. Changes ................................................................ 20
   3.5. Solved Issues .......................................................... 20
4. Known Issues ................................................................. 22
1. Release Summary

Release benefits

*Introducing the Neo Licensing System, a new, flexible licensing system*

The *Neo Licensing System* is the new licensing system from Euresys. It is now available in *Open eVision* in addition to the existing licensing systems, which remain available.

- The *Neo Licensing System* is reliable, easy to use and based on state-of-the-art technology.
- Choose to activate your license on a *Neo USB Dongle* or in a *Software Container*. Buy a license and decide later.
  - The *Neo USB Dongles* offer a sturdy hardware and the flexibility to be transferred easily from a computer to another.
  - The *Software Containers* do not need any dedicated hardware. They are linked to the computer on which you activate them.

- The *Neo License Manager* provides two interfaces:
  - An intuitive, easy to use, *Graphical User Interface*.
  - A *Command Line Interface* for easy automation of the licensing procedures.
**EasySegment Supervised**

- In addition to the existing *unsupervised segmentation* mode, *supervised segmentation* is now available in EasySegment.

- When using the *supervised* mode, first annotate the pixels of defects or of objects in your images to train the EasySegment Supervised tool. The tool then learns to detect these defects or objects in new images.

- Annotate the pixels of defects or of objects in your images to train the EasySegment Supervised tool. The tool then learns to detect these defects or objects in new images.

<table>
<thead>
<tr>
<th>Supervised mode</th>
<th>Unsupervised mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>The <em>supervised</em> mode directly builds a model of the defects from the annotated images.</td>
<td>The <em>unsupervised</em> mode does not require you to annotate pixels of defects.</td>
</tr>
<tr>
<td>The <em>supervised</em> mode can detect and segment more types of defects with better accuracy than the <em>unsupervised</em> mode.</td>
<td>The <em>unsupervised</em> mode automatically builds a model of the good images and tries to detect variations from this model.</td>
</tr>
<tr>
<td>You can also use the <em>supervised</em> mode to segment other types of objects than defects.</td>
<td></td>
</tr>
</tbody>
</table>

- In Deep Learning Studio:
  - A new interactive pixel annotation tool is now available:
The simple tool configuration:

The various new metrics for evaluation (pixel, blob and image based metrics):

The result visualization:

In the API:
- The dataset supports many new methods for the EClassificationDataset class.
- The supervised segmentation tool supports the new classes:
  - ESupervisedSegmenter for the segmentation tool.
  - ESupervisedSegmenterResult for the segmentation result.
  - ESupervisedSegmenterMetrics for various metrics about the tool.
EasyQRCode

- The code recognition and reading performance of **EasyQRCode** has been significantly improved in this release.

- The speed of the Read method exhibits an average gain of 50% compared to previous versions. This value can be much greater on some images, particularly when multiple codes are present in the image.

- The images of heavily distorted codes, with non-uniform cell sizes are now more reliably located and decoded thanks to an improved grid positioning algorithm.

- Severely overprinted codes are better handled thanks to new digitization recovery algorithms.

Results without (left - could not decode) and with (right - decoding successful) the additional digitization algorithms (red squares are wrongly digitized cells)

Open eVision Studio

- **Open eVision Studio** now displays non-latin characters.
2. Release Specifications

OS and processor architectures

- Open eVision is a 32-bit and 64-bit library that requires a processor compatible with the SSE2 instruction set.

- Open eVision runs on the following operating systems:

<table>
<thead>
<tr>
<th>OS version</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 10®</td>
<td>32-bit</td>
</tr>
<tr>
<td>Windows 10®</td>
<td>64-bit</td>
</tr>
<tr>
<td>Windows 8®</td>
<td>32-bit</td>
</tr>
<tr>
<td>Windows 8®</td>
<td>64-bit</td>
</tr>
<tr>
<td>Windows 7®</td>
<td>32-bit</td>
</tr>
<tr>
<td>Windows 7®</td>
<td>64-bit</td>
</tr>
</tbody>
</table>

- Remote connections:
  - You can install and use Open eVision licenses on a remote connection using remote desktop, TeamViewer or any other similar software.

- Virtual machines:
  - You cannot install Open eVision on virtual machines.

Supported IDE and programming languages

Select the recommended API Module according to your IDE and programming language:

<table>
<thead>
<tr>
<th>IDE</th>
<th>C++</th>
<th>C#, VB.NET, C++/CLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Visual Studio 2008® SP1</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2010®</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2012®</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2013® (*)</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2015®</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2017®</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2019® (**)</td>
<td>C++</td>
<td>.NET Assembly</td>
</tr>
</tbody>
</table>

**NOTE**

(*) Visual C++ MFC MBCS Library for Visual Studio 2013 must be installed.

(**) In Visual Studio 2019, to be compatible with Open eVision, the C++ Language Standard option must be left empty or set to ISO C++14 Standard.
Required system resources

- Display size:
  - Minimum: 800 x 600
  - Recommended: 1280 x 1024

- Display color depth:
  - Minimum: 16 bits
  - Recommended: 32 bits

- Hard disk space:
  - Open eVision libraries: 100 MB - 2 GB (depending on selected options)
3. Release Details

3.1. Updated Development Tools Support

Support Visual Studio 2019

- **Visual Studio 2019** is supported from release 2.13.

**NOTE**
To be compatible with **Open eVision**, the **C++ Language Standard** option must be left empty or set to **ISO C++14 Standard**.

3.2. New and Improved Features

**New features**

*Introducing the Neo Licensing System, a new, flexible licensing system*

The **Neo Licensing System** is the new licensing system from Euresys. It is now available in **Open eVision** in addition to the existing licensing systems, which remain available.

- The **Neo Licensing System** is reliable, easy to use and based on state-of-the-art technology.
- Starting now, it is available under Windows.
- Choose to activate your license on a **Neo USB Dongle** or in a **Software Container**. Buy a license and decide later.
  - The **Neo USB Dongles** offer a sturdy hardware and the flexibility to be transferred easily from a computer to another.
  - The **Software Containers** do not need any dedicated hardware. They are linked to the computer on which you activate them.
The Neo License Manager provides two interfaces:
- An intuitive, easy to use, **Graphical User Interface**.
- A **Command Line Interface** for easy automation of the licensing procedures.

- **Current dongle- and software-based licensing systems:**
  - The current licensing systems remain fully supported.
  - If you are using one of these, activate and use your licenses just as before.
  - The License Manager for your licenses is still available at its usual location.

- To avoid issues with **multiple licensing systems**, such as a longer startup time:
  - Select the one(s) you want to use with the new
    `Preconfiguration::SelectLicensingModels` method.
  - By default, all supported licensing systems are enabled.

**EasyDeepLearning**

**EasyDeepLearning** now supports unlabeled images in the dataset.

- You can use these unlabeled images for testing, but not for training or validation.
- You can import all your images into your dataset without annotating them before the training step.
EasySegment Supervised

- In addition to the existing unsupervised segmentation mode, supervised segmentation is now available in EasySegment.

- When using the supervised mode, first annotate the pixels of defects or of objects in your images to train the EasySegment Supervised tool. The tool then learns to detect these defects or objects in new images.

- Annotate the pixels of defects or of objects in your images to train the EasySegment Supervised tool. The tool then learns to detect these defects or objects in new images.

<table>
<thead>
<tr>
<th>Supervised mode</th>
<th>Unsupervised mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>The supervised mode directly builds a model of the defects from the annotated images.</td>
<td>The unsupervised mode does not require you to annotate pixels of defects.</td>
</tr>
<tr>
<td>The supervised mode can detect and segment more types of defects with better accuracy than the unsupervised mode.</td>
<td>The unsupervised mode automatically builds a model of the good images and tries to detect variations from this model.</td>
</tr>
<tr>
<td>You can also use the supervised mode to segment other types of objects than defects.</td>
<td></td>
</tr>
</tbody>
</table>

- In Deep Learning Studio:
  - A new interactive pixel annotation tool is now available:
- The simple tool configuration:

- The various new metrics for evaluation (pixel, blob and image based metrics):

- The result visualization:

- In the API:
  - The dataset supports the following new methods for the `EClassificationDataset` class:
    - `SetSegmentationMap`, `GetSegmentationMap`
    - `UnsetSegmentation`, `ResetSegmentation`
    - `AddRegionToSegment`, `GetRegionForSegment`
    - `GetNumSegmentationLabels`
    - `GetNumImagesForSegmentationLabel`, `GetNumPixelsForSegmentationLabel`
- HasSegmentation, HasForegroundSegments
- GetNumSegmentedBlobs
- GetSegmentationLabel, SetSegmentationLabel
- AddSegmentationLabel, RemoveSegmentationLabel
- GetSegmentationLabelWeight, SetSegmentationLabelWeight
- GetNumImagesWithSegmentation, GetNumImagesWithoutSegmentation
- GetNumImagesWithForegroundSegments
- GetNumImagesWithoutForegroundSegments

- The supervised segmentation tool supports the following new classes:
  - ESupervisedSegmenter for the segmentation tool.
  - ESupervisedSegmenterResult for the segmentation result.
  - ESupervisedSegmenterMetrics for various metrics about the tool.

**Easy3D**

- Use the new `ClearRenderSource` method in `E3DViewer` to remove all the displayed entities.

- To use the new point picking feature:
  - In the `3DViewer`, CTRL + left click to select the point closest to the mouse position.
  - Use the new methods of the `E3DViewer` class:
    - Pick3DPoint
    - SetPickingDistanceThreshold
    - SetPickingDisplay, SetPickingLabelSize, SetPickingLabelColor.

- To find the `E3DPoint` closest to a segment in an `EPointCloud`, use the `EPointCloud::DistanceToSegment` method.

- If you enable the multicore processing (see `Easy::SetMaxNumberOfProcessingThreads`), the expected speed up is:
  - 2 threads: 1.96
  - 3 threads: 2.88
  - 4 threads: 3.62
• You can use the new space partition optimizations for the point cloud.
  □ Use EPointCloud::SetEnableSpacePartition to activate the 3D space partitioning and speed up the geometric queries such as EPointCloud::DistanceToSegment.
  □ Comparison of EPointCloud::DistanceToSegment execution time in various contexts:

<table>
<thead>
<tr>
<th>Number of points:</th>
<th>100 000</th>
<th>1 000 000</th>
<th>10 000 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query without the space partition</td>
<td>1.5 ms</td>
<td>15.7 ms</td>
<td>160 ms</td>
</tr>
<tr>
<td>Building the space partition</td>
<td>3 ms</td>
<td>53 ms</td>
<td>390 ms</td>
</tr>
<tr>
<td>Query with the space partition</td>
<td>0.06 ms</td>
<td>0.11 ms</td>
<td>0.31 ms</td>
</tr>
</tbody>
</table>

• Easy3DObject

□ To tune the extraction of the objects, adjust the sensitivity with the new E3DObjectExtractor::SetExtractionSensitivity method.
  □ With a high sensitivity, the tool tries to extract objects mixed with their surrounding.
  □ With a low sensitivity, the tool tends to ignore faint objects.

Extraction sensitivity = 0.5 (top) and 0.8 (bottom)
To detect and enhance the contours or frontiers between objects, use the new \texttt{E3DObjectExtractor::SetContourReinforce} method. It is mainly useful when objects to extract are in contact with one another.

![Contour reinforcement OFF (left) and ON (right)](image)

- The demo application \texttt{3DObjectExtraction} is updated and you can now resize the window.

\textit{Open eVision Studio}

- \textbf{Open eVision Studio} now displays non-latin characters.

\textbf{Improvements}

\textit{Easy3D}

- The new \texttt{EPointCloud::LoadPLY} and \texttt{EPointCloud::SavePLY} methods support the PLY binary file format.
  - This format is documented here: http://paulbourke.net/dataformats/ply/.
- The \texttt{EPointCloud} load methods now support an optional filter to discard unwanted points.
**EasyColor**

- The Bayer demosaicing has been improved in quality and speed.
  - The BayerToC24 method, which converts Bayer raw images to RGB images, has been improved.
  - It features new interpolation modes and faster conversion.
  - With the new modes, the resulting color image shows less artifacts (zipper effect and color aliasing), as illustrated:

- The new modes are available in **Open eVision Studio**, in the **Color Conversion** window.
**EasyQRCode**

- The code recognition and reading performance of **EasyQRCode** has been significantly improved in this release.
- The speed of the **Read** method exhibits an average gain of 50% compared to previous versions. This value can be much greater on some images, particularly when multiple codes are present in the image.
- The images of heavily distorted codes, with non-uniform cell sizes are now more reliably located and decoded thanks to an improved grid positioning algorithm.
- Severely overprinted codes are better handled thanks to new digitization recovery algorithms.

![Results without (left - could not decode) and with (right - decoding successful) the additional digitization algorithms (red squares are wrongly digitized cells)](image)

**Open eVision Studio**

- The **EasyQRCode** tool interface is significantly improved.
Deep Learning Studio

- The Deep Learning Studio logs in Memento now provide various debugging information related to the hardware configuration.
- Deep Learning Studio better detects the GPU installation and displays the adequate messages.

In Deep Learning Studio, you can now continue to train a previously trained deep learning tool.

EasySegment

- The training algorithm of the Unsupervised Segmentation tool has been updated for better and more consistent results.

General

- C# samples now use and illustrate the best practices for the memory management.

EasyOCR

- The documentation of the EasyOCR parameters related to the minimal character width and the character spacing have been updated to be more comprehensive.
3.3. Breaking Changes

Starting with this release 2.13, Open eVision implements the following changes:

**EasyDeepLearning**

- The `EClassificationDataset::SetGlobalRegionOfInterest` and the `EClassificationDataset::SetGlobalMask` classes are removed.
  - Starting from release 2.13.0, to use a common ROI and/or mask on the images of the dataset, simply set the same ROI and/or mask to all the images of the dataset.
  - In **Deep Learning Studio**, use the new button **Apply to All Images** to set the same ROI and/or mask to all the images in the dataset.

3.4. Changes

Starting with this release 2.13, Open eVision implements the following changes:

**EasyDeepLearning**

- The `EUnsupervisedSegmenterMetrics` class is modified to be a descendant of the new `EDeepLearningDefectDetectionMetrics` class.
  - The API is not changed.

**EasyQRCode**

- The `Detect` and `Decode` methods are deprecated.
  - Starting with release 2.13, no documentation is provided anymore for these methods.
- The default value of the `CellPolarityConfidenceThreshold` parameter is set to `0.2f`.
  - This parameter can improve the reading rate of overprinted QR codes.

3.5. Solved Issues

The following issues have been fixed in Open eVision 2.13:

**EasySegment**

- (2.13.0) `EUnsupervisedSegmenterResult::GetRegion` now returns the detected defective region instead of the entire image.
Open eVision Studio

- (2.13.0) In EasyQRCode, the crash that occurred when displaying information of unreliably detected QR codes in ECI mode is fixed.
- (2.13.0) In EasyQRCode, the detection trade-off is now correctly reset to the indicated value when you uncheck the custom trade-off box.
- (2.13.0) In EasyOCR, the preview of the segmentation is now correct. The pixels equal to the threshold are now correctly excluded.

EasyQRCode

- (2.13.0) The PerspectiveLegacy detection method does not throw the "Parameter 1 out of range" or the "Invalid operation - See Reference" EExceptions anymore.

Easy3D

- (2.13.0) The 3DCoaxlinkLLE sample program now correctly opens the folder selector to save the captured images.
- (2.13.0) The 3DProcessing3D sample program now comes with a correct EasyOCR2 model for the 16-bit depth maps.

EasyFind

- (2.13.0) The assignment operator of EPatternFinder class does not cause a heap memory corruption anymore as it did in rare circumstances.

ERegion

- (2.13.0) ERegion::CropRuns now returns the correct bounding box of the region when the returned region is empty.
- (2.13.0) ERegion::CropRuns now always returns the correct region when the cropped Y origin is different from 0.

EasyOCR2

- (2.13.0) In EasyOCR2, the detection of a topology with a word that contains zero character does not fail anymore for a fixed detection method.
4. Known Issues

Licensing

On some installations, the licensing systems can take a long time to start (from 10 seconds up to a few minutes). If you have this issue, you can try the following procedures:

- Clean your software license cache.
  - The software license cache can become bloated by usage.
  - It can also happen if you use only dongles, as the system checks the presence of software licenses in all cases.
  - To clean the cache, use the `LicenseManager.exe /DeleteLicenseFiles` command.

**WARNING**
This command deletes all the licenses that are not managed by the Neo License Manager on the system. Reactivate these licenses after the cleaning.

- Update your system root certificates.
  - If your root certificates are expired, the validation of the licensing system signatures might fail and timeout.
  - This only happens if the computer is on a network, even if the network is not connected to the Internet.

- Enable only the licensing system(s) you use.
  - By default, all the supported licensing systems are enabled.
  - Use the new (available from 2.13) `Preconfiguration::SelectLicensingModels` method to select exactly the licenses you want to enable and avoid issues arising from the usage of the other ones.

**.NET API and unsigned integer parameters**

Since this release 2.5 of Open eVision, unsigned integer parameters in the C++ API are not exposed in the .NET API as signed integer anymore, but as unsigned integers. This brings the .NET API closer to the C++ one.

This change does not cause any issue except when you want to pass an enumerate value as one of these parameters. In these specific cases, update your casting operation as in the following example:

```csharp
codedImage.SetThreshold((int)EThresholdMode.MinResidue);
```

becomes:

```csharp
codedImage.SetThreshold((unchecked(uint)EThresholdMode.MinResidue));
```
Reserved keywords

- The following keywords are reserved by Open eVision:
  - EUnit_um, EUnit_mm, EUnit_cm, EUnit_dm
  - EUnit_m, EUnit_dam, EUnit_hm, EUnit_km
  - EUnit_mil, EUnit_inch, EUnit_foot, EUnit_yard, EUnit_mile
  - EasyWorld

**TIP**
To avoid conflict, do not use these keywords to name variables, functions, methods, macros...

Image formats

- If you use some types of 96-bit RGB Tiff image, Open eVision may crash.

Memory leaks

- If you use the CRT library to detect memory leaks in your program, it can falsely detect some memory leaks when you use the Open eVision library.
  - This is a known limitation of the CRT library memory leak detection scheme.
  - It happens when the memory leak detection scheme is ended before the Open eVision DLL is unloaded or the code in the Open eVision headers is uninitialized.

Object cleanup: .NET

As a rule, it is highly recommended to call `Dispose()` on Open eVision .NET objects when they are not useful anymore.

**TIP**
Not doing so might result in unnecessarily high memory usage and crashes.

Example in C#

```csharp
using(EImageBW8 src = new EImageBW8())
using(EPatternFinder finder = new EPatternFinder())
{
    src.Load(ImageFilePath);
    EFoundPattern[] foundPatterns = finder.Find(src);
    ...
    foreach(EFoundPattern foundPattern in foundPatterns)
    {
        foundPattern.Dispose();
    }
}
```

In addition, if you use a nested object (such as the segmenter properties in EasyObject encoder objects), remember to call `Dispose()` on that object before calling `Dispose()` on the parent object.
Example in C#

```csharp
imageEncoder.GrayscaleSingleThresholdSegmenter.BlackLayerEncoded = true;
...
imageEncoder.GrayscaleSingleThresholdSegmenter.Dispose();
imageEncoder.Dispose();
```

Basic types: retrieving and setting pixel values

Using the `GetPixel()` and `SetPixel()` methods of the various ROI classes can sometimes be slow if you make many calls (regardless of the language used).

- In order to greatly speed up the ROI/image buffer access, embed the buffer access in your own code.
- See the examples below that use the new Open eVision API.

**NOTE**
For a better readability of these examples, the variable declarations and initializations have been omitted when possible.

Example in C++

```c++
void* pixAddr;
UINT8 pix;
...
for (int y = 0; y < height; ++y)
{
    pixAddr = bw8Image.GetImagePtr(0,y);
    for (int x = 0; x < width; ++x)
    {
        pix = *(reinterpret_cast<UINT8*>(pixAddr)+x);
    }
}
```

Example in C#

```csharp
using System.Runtime.InteropServices;
...
IntPtr pixAddr;
byte pix;
...
for (int y = 0; y < height; ++y)
{
    pixAddr = bw8Image.GetImagePtr(0,y);
    for (int x = 0; x < width; ++x)
    {
        pix = Marshal.ReadByte(pixAddr,x);
    }
}
Basic types: ROI zooming and panning issue

- When drawing an ROI with a zoom factor, applying panning (retrieved from a scroll bar) causes the ROI display to be shifted. Consequently, the HitTest() and Drag() functions fail because the handles do not appear at their actual positions.

  Workaround: The panning values should be divided by the zoom factor before calling the DrawFrame(), HitTest() and Drag() functions.

Basic Types: miscellaneous issues

- TIFF files containing RGB values + alpha values are not supported.
- Filenames with multibyte characters are not supported. The error is "Unrecognized file format".
- Easy::GetBestMatchingImageType() only works for BW8 and C24 images.

Multithreading

- In multithread applications, if Easy::Initialize is not called before launching new threads that call Open eVision functions, then the number of Open eVision processing threads in these new threads may be wrongly initialized to use all the cores that are available on the machine.

EasyBarCode

- Due to a bug in the debugger of Visual C++ 2012, the reading time of bar codes may increase after a failed reading. This happens only in debug mode with Visual C++ 2012.
- EasyBarCode requires that a quiet zone of at least one full module is present around the whole bar code to be read.
- EasyBarCode is currently unable to read bar codes with curved or distorted bars. For reliable reading, the bars must be as straight as possible.
- EasyBarCode is currently not multithread-safe.

EasyOCR2

- (2.13.0) The detection of a topology with ranged characters was always failing with the proportionnal detection method. This functionnality is now disabled and an error is thrown.

EasyQRCode

- EasyQRCode does not support MicroQR codes.

EasyObject

- The ECodedImage2 and EHarrisDetector results are drawn slowly when there are many results.
EasyMatch

- By design, the maximum size for a pattern in EasyMatch is 1791 x 1791.
- Matching a vertically symmetric pattern with an angle tolerance around 180° and in the original image can lead to an error of 1 pixel on the detected position.
- By default, EasyMatch interpolation does not work on 15 x 15 and smaller patterns.

  **Workaround:** For pattern sizes smaller than 16 x 16, adjust the MinReduced area to fit the MinReducedArea < W*H/4 (if interpolation is needed).

EasyGauge

- In .NET, the EPointGauge.GetMeasuredPoint() overload with no argument is not available. To get the default measured point, use -1 as index.
- By design, an ELineGauge, ERectangleGauge, ECircleGauge or EWedgeGauge is reported as invalid if at least one of its sample points is invalid. In addition, these invalid sample points cannot be drawn as they have not been measured successfully.
- The EWedgeGauge::SetActiveEdges() method incorrectly gets the EDragHandle_EDGE_R and EDragHandle_EDGE_RR bits mixed up when processing its argument.

  **Workaround:** In order to activate the inner circle, set the EDragHandle_EDGE_RR flag and use the EDragHandle_EDGE_R flag to activate the outer circle.

- Using a gauge on an ROI leads to drawing problems.
  
  **Workaround:** Use the gauge on the parent image.

- In the custom EDraggingMode_ToEdges dragging mode, you cannot resize the nominal wedge gauge position using the on-screen handles, neither in a custom application nor in Open eVision Studio or in Open eVision Eval.

  **Workaround:** Enter numerical values for the wedge gauge position.

EasyMatrixCode

- When grading is enabled, the optimizations are made in order to get accurate grading rather than have the best possible reading. As a result, the number of decoding errors reported with grading can be higher than without grading.
- Inspecting images with a lot of details, even if they are low contrast, can require much more time spent in EasyMatrixCode than the TimeOut set previously.
- In .NET, retrieving the coordinates of a MatrixCode using EMatrixCode.GetCorner() or EMatrixCode.Center() can lead to an unhandled exception when the garbage collection starts up. To avoid this problem, call Dispose() on the EPoint objects returned by these functions when they are no longer needed.

Easy3DObject

- The E3DObjectExtractor objects saved with Open eVision 2.11 cannot be loaded with Open eVision 2.12.
Open eVision Studio

- In the ROI management dialog, clicking on a ROI in the tree view does not activate the ROI overlay in the image window. This can prevent you to graphically interact with it.

  To avoid this issue and to properly interact with the ROI overlay:
  a. Click on the ROI in the tree view.
  b. Immediately after, click inside its overlay in the image window.

- To avoid crashes, deselecting all detection methods in the EasyQRCode dialog box reverts to the default detection method. In some cases, the dialog might not refresh automatically.

- In the detection method selection control of the EasyQRCode dialog box, clicking beside a text might select or deselect it.

- When managing the EasyOCR2 topology, the potential characters option is not available.

Open eVision installer

- There is a conflict between the Open eVision installer and any program using the UDP:6001 port. When a software is already using this port, the installation fails and rolls back.

  Workaround: Install Open eVision first, and then the other software.

  **NOTE**
  This port is typically used by National Instrument software such as LabView.

- Before installing any Euresys product, make sure that your OS is up-to-date (using Microsoft Update), otherwise, problems might occur.

Open eVision License Manager

- Under Windows XP, the Open eVision License Manager might not start if the .NET Framework 2.0 is not installed.

- Using the Open eVision License Manager to activate a license requires an Internet connection and a secure SSL transaction to EURESYS s.a. servers.

  **NOTE**
  On older systems, such as Windows XP SP3, ensure that the root certificates are up-to-date otherwise the secure connection is refused and the license is not activated.
When activating an emergency license, the following error may occur: “Error Message: Loading of the ASR failed!”

This error occurs when all 3 emergency licenses have already been used and the computer has been formatted.

Using Open eVision License Manager in English language mode on a Chinese or Japanese Windows version can lead to truncated text being displayed. This is an issue linked to the automatic font selection and there is currently no workaround. Please note however that, by default, the Open eVision License Manager runs in the OS language, including Chinese and Japanese.