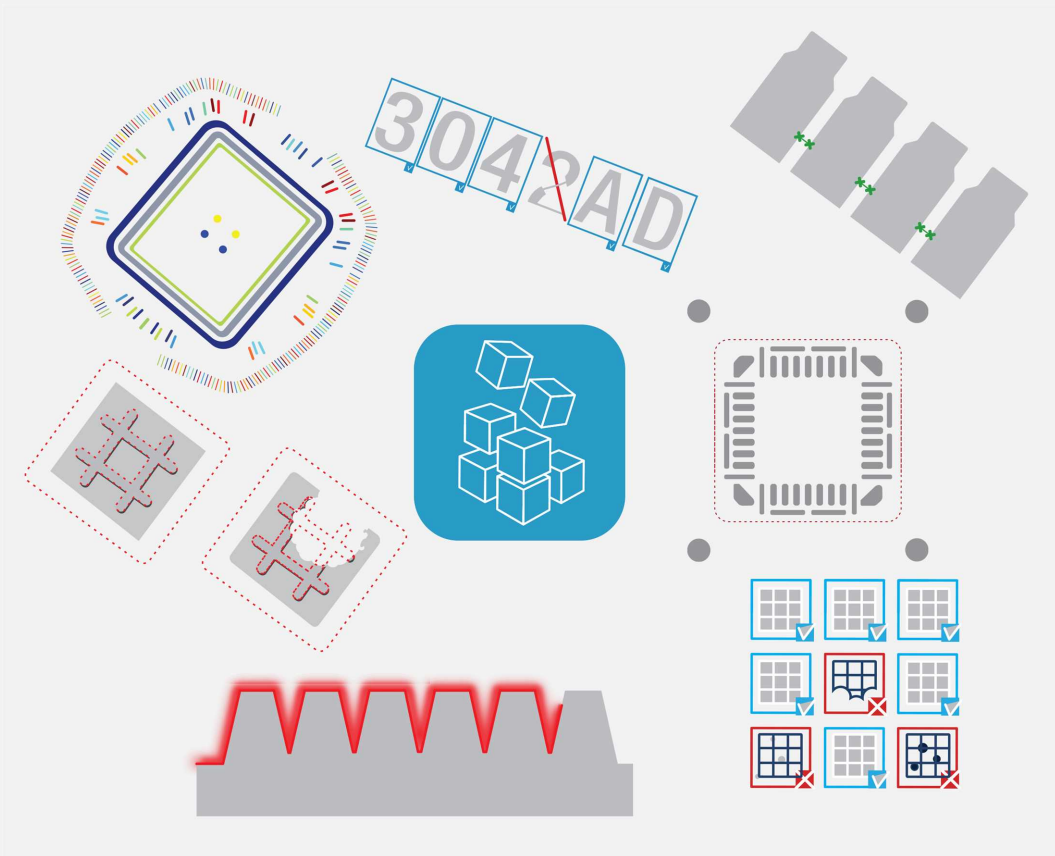


# Open eVision

Release 2.16.1



This documentation is provided with Open eVision 2.16.1 (doc build 1155).  
[www.euresys.com](http://www.euresys.com)

# Contents

1. Release Benefits .....	4
2. Release Specifications .....	6
3. Release Details .....	8
3.1. New and Improved Features .....	8
3.2. Breaking Changes .....	13
3.3. Changes .....	14
3.4. Solved Issues .....	16
4. Known Issues .....	18

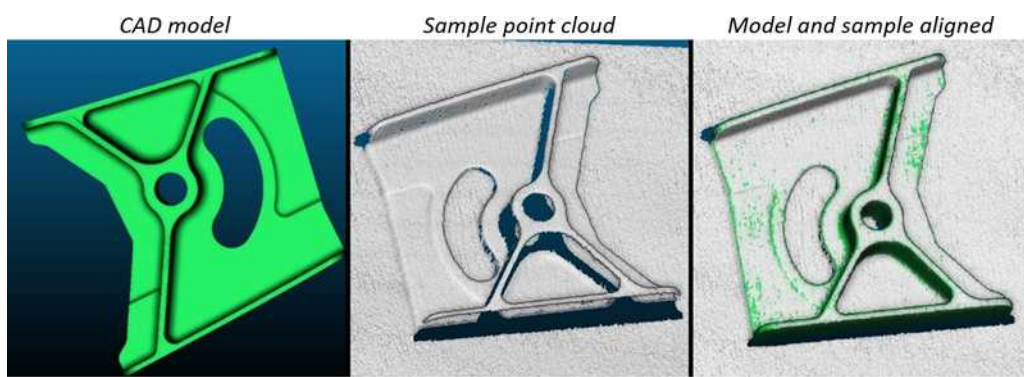
# 1. Release Benefits

Starting with this release 2.16, Open eVision offers the following new products and functions:

## Easy3DMatch

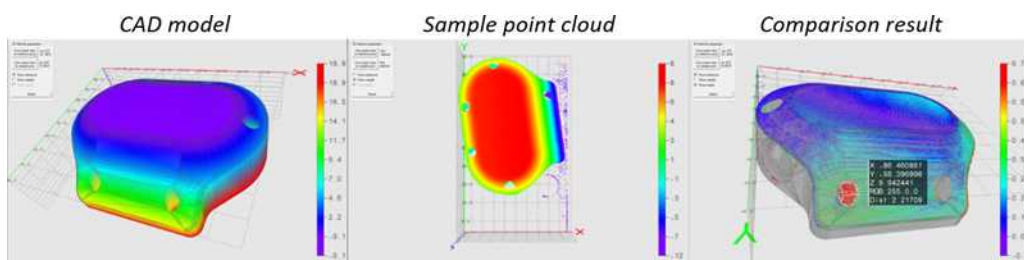
**Easy3DMatch** is a new Open eVision tool.

- It features alignment functions to find the exact pose (position and orientation) of acquired 3D objects using a reference model. This model can be specified as a reference point cloud or as a 3D mesh from CAD software.



(3D CAD models courtesy of Direct Dimensions)

- After the alignment of the 3D data with the reference model, **Easy3DMatch** can search for local discrepancies, like holes, bumps, scratches, gaps... The list of detected anomalies is returned along with their individual position, distance and area. New point clouds with distances and colored data can be generated and displayed by the interactive Open eVision 3D viewer. **Easy3D** is required for all 3D developments and is included in the **Easy3DMatch** license.



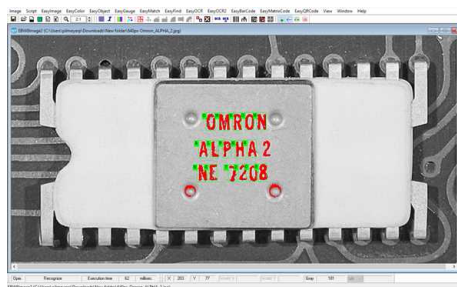
(3D CAD models courtesy of Direct Dimensions)

- **Easy3DMatch** is compatible with all 3D sensors that produce point clouds, depth maps or height maps (ZMap).



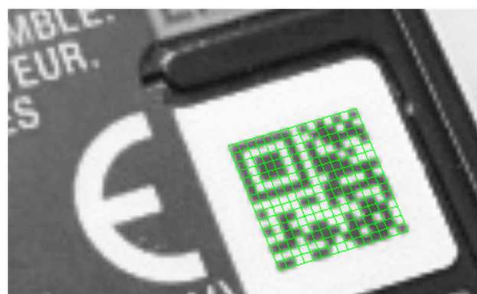
### EasyOCR2

- **EasyOCR2** now supports Optical Character Recognition powered by deep learning technologies. It comes with two pre-trained character classifiers that work out of the box and do not require any training! Read short texts such as part numbers, serial numbers or date codes printed using standard industrial fonts or the OCR-A font. Both classifiers support uppercase letters, numbers and the most common punctuation marks. No GPU is required. The **OCR2Demo** sample program as well as Open eVision Studio have been updated to support the new functionality.



### EasyQRCode

- **EasyQRCode** now supports Micro QR codes. Micro QR codes are a more compact version of the regular QR codes used when less data needs to be encoded.



### Linux support

- Open eVision now runs on Linux: Linux 64 bits (x86-64) operating systems with a glibc version greater or equal to 2.18.

## 2. Release Specifications

### OS and processor architectures

- Open eVision is a 32-bit and 64-bit library that requires a processor compatible with the SSE4 instruction set.
- Open eVision runs on the following Windows operating systems:

OS version	Additional information	
Windows 10®	32-bit	—
Windows 10®	64-bit	—
Windows 8®	32-bit	—
Windows 8®	64-bit	—
Windows 7®	32-bit	The recommended version is 6.1.7601 (Windows 7 Service Pack 1)
Windows 7®	64-bit	

- Open eVision runs on the following Linux operating systems:
  - Linux 64 bits (x86-64) with a glibc version greater or equal to 2.18.
  - This release has been validated with the following distributions:
    - Ubuntu 16.04 LTS
    - Ubuntu 18.04 LTS
    - Ubuntu 20.04 LTS
- 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
  - Linux virtual machines are supported. Microsoft Hyper-V and Oracle VirtualBox hypervisors have been successfully tested.
  - You cannot install Open eVision on Windows virtual machines.

### Supported IDE and programming languages

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

IDE	Programming language	
	C++	C#, VB.NET, C++/CLI
Microsoft Visual Studio 2008® SP1	C++	.NET Assembly
Microsoft Visual Studio 2010®	C++	.NET Assembly
Microsoft Visual Studio 2012®	C++	.NET Assembly
Microsoft Visual Studio 2013® (*)	C++	.NET Assembly
Microsoft Visual Studio 2015®	C++	.NET Assembly
Microsoft Visual Studio 2017®	C++	.NET Assembly

IDE	Programming language	
	C++	C#, VB.NET, C++/CLI
Microsoft Visual Studio 2019®	C++	.NET Assembly
QtCreator 4.15 with Qt 5.12 (**)	C++	

**NOTE**

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

(\*\*) A C++ compiler like **MSVC** must be installed

## 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

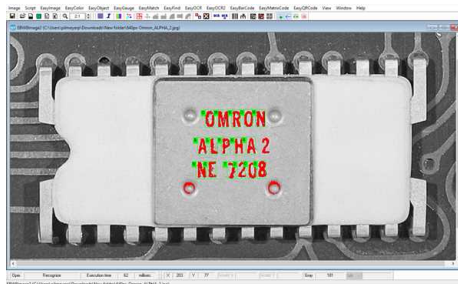
### 3.1. New and Improved Features

#### New features

---

##### *EasyOCR2*

- **EasyOCR2** now supports Optical Character Recognition powered by deep learning technologies. It comes with two pre-trained character classifiers that work out of the box and do not require any training! Read short texts such as part numbers, serial numbers or date codes printed using standard industrial fonts or the OCR-A font. Both classifiers support uppercase letters, numbers and the most common punctuation marks. No GPU is required. The **OCR2Demo** sample program as well as Open eVision Studio have been updated to support the new functionality.

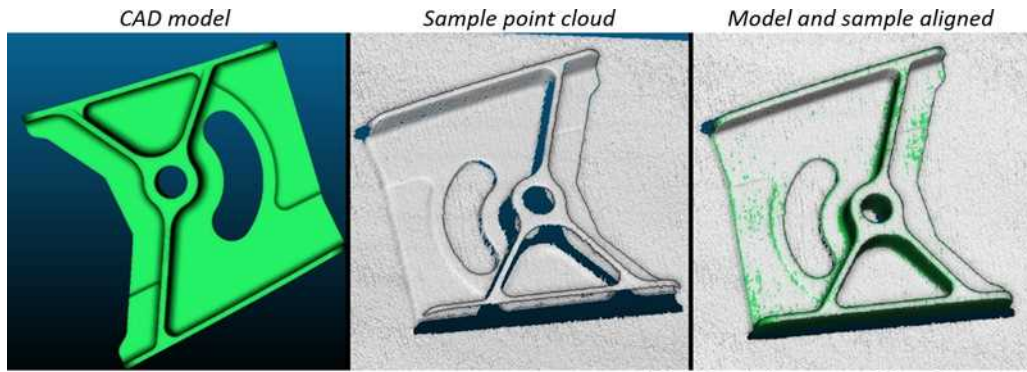




### Easy3DMatch

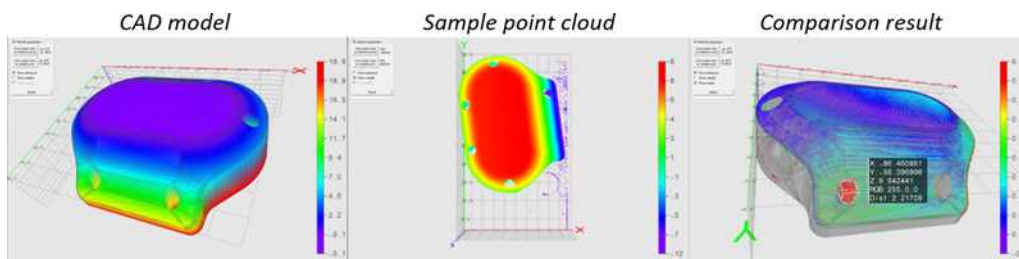
**Easy3DMatch** is a new Open eVision tool.

- It features alignment functions to find the exact pose (position and orientation) of acquired 3D objects using a reference model. This model can be specified as a reference point cloud or as a 3D mesh from CAD software.



(3D CAD models courtesy of Direct Dimensions)

- After the alignment of the 3D data with the reference model, **Easy3DMatch** can search for local discrepancies, like holes, bumps, scratches, gaps... The list of detected anomalies is returned along with their individual position, distance and area. New point clouds with distances and colored data can be generated and displayed by the interactive Open eVision 3D viewer. **Easy3D** is required for all 3D developments and is included in the **Easy3DMatch** license.



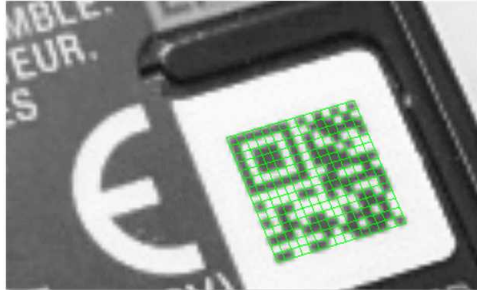
(3D CAD models courtesy of Direct Dimensions)

- **Easy3DMatch** is compatible with all 3D sensors that produce point clouds, depth maps or height maps (ZMap).



## EasyQRCode

- **EasyQRCode** now supports Micro QR codes. Micro QR codes are a more compact version of the regular QR codes used when less data needs to be encoded.



- You can serialize an `EasyQRCodeReader` object.

## Easy3D

- **Easy3DGrab: Zivid**, a new 3D sensor is added to the compatibility list.
- `EnableSmartColorGeneration`, a new function in `EColorRamp`, allows you to discard or not the outliers for the generations of colors (previously the outliers were always discarded).
- `EGridDecimator`, a new `EDecimator`, performs decimation by defining a regular grid in 3D space and transforms all the points contained in a grid cell to their centroid.
- `E3DPlane` offers two new methods:
  - `IntersectionWithTwoPlanes` to compute the intersection between three planes.
  - `AngleWithPlane`, to compute the angle between two planes.
- `E3DViewer` now supports multiple render sources.
  - A render source can be a point cloud, a mesh or a ZMap.
  - You can display each render source with specific attributes, like colors and opacity.
  - The `E3DViewer` sample now exposes these new features. The `E3DViewer` class and sample implement rapid change in view direction with keys X, Y and Z. The picking action (CTRL + LEFT CLICK) now also displays the picked point attributes.
- **Easy**
  - The methods `SetOemKey` and `CheckOemKey` now support the 6514 Neo USB Dongles in addition to the 6512 USB Dongle. To use an OEM Key with a Neo USB Dongle, you must prepare that dongle with the Neo License Manager (see below).
- Neo License Manager
  - The Neo License Manager can now prepare 6514 Neo USB Dongles for the OEM Key functionality. When a preparation is required, a warning text appears in the list of the dongle licenses. Click on that text and follow the instructions.

## Improvements

---

### Easy

- The new method `EFloatRange::Update(EFloatRange&)` updates the range depending of another range.
- The new method `EFloatRange::IsValid` returns `TRUE` if the range is defined and valid

### Easy3D

- The new method `EPointCloud::HasAttributeBuffer` checks if an attribute is present.
- The new methods `EPointCloud::AllocateAttributeBuffer` and `EPointCloud::AllocateCustomAttributeBuffer` create an attribute buffer without precomputed data.
- The methods `EPointCloud::Save(const std::string path)`, `EPointCloud::Load(const std::string path)` save and load the point cloud based on the extension (Open eVision proprietary, CSV, OBJ, PCD, PLY and XYZ).
- The methods `EMesh::Save(const std::string path)` and `EMesh::Load(const std::string path)` save and load the mesh based on the extension (Open eVision proprietary, STL).
- The new method `EMesh::Clear` removes all triangles from an `EMesh`.
- The method `EAffineTransformer` is now around 4 times faster when transforming point clouds if the source and destination `EPointCloud` parameters are different.
- The speed of reading and writing ASCII STL files (`EMesh`) has been improved and both operations are now around two times faster. Reading and writing binary STL files are supported and are around 40 times faster than the old ASCII reading and writing.
- The method `EObjectBasedCalibrationGenerator::Compute` is now around 4 times faster due to internal reworking of the method. As a side effect, small differences in the calibration result may occur.

### Qt samples

- The Qt Draw Adapter now supports the rotated text.

### EasyOCR2

- The drawing of recognition has been improved when using variable space fonts.
- **OCR2Demo** is updated and is now able to adjust all `EOCR2` parameters.
- The process of resizing character for recognition has been accelerated.

### EasyObject

- The speed of the calculation of the Minimum Enclosing Rectangle has been significantly improved.

### EasyQRCode

- (2.16.1) The method `EQRCode Draw` has been improved.

### EasyClassify

- The colored heat maps are now available.
- (Breaking change) The method `EClassifier::GetHeatMap` now returns an `EImageC24` instead of an `EImageBW8`.
- For networks with a capacity of `EClassifierCapacity_Small`, the minimum width and height are now 64.

### EasySegment Unsupervised

- The post-processing of `EasySegment Unsupervised` results has been improved to make it faster. The GPU speed improvements are between 7% and 50% according to the batch size.
- (Breaking change) There can be small differences in the results compared to previous versions (score, segmented regions...).

### EasySegment Unsupervised /Supervised

- The new batch inference methods, that take pointers to images and results, make inference faster (1% to 30% faster according to batch size and parametrization) as they require less object copies.

### Deep Learning Studio

- The processing time of an image does not include the time required to load the image from the hard drive anymore.

### Open eVision Studio

- **EasyOCR2** Topology UI has been updated to allow potential characters.
- (2.16.1) In the **EasyBarCode** tool, the selection of the symbologies has been improved.

### Neo License Manager

- The Neo License Manager now asks you for a confirmation when you try to apply a Ticket ID containing a license to a container that already has that license.

## 3.2. Breaking Changes

Starting with this release 2.16, Open eVision implements the following changes:

### *EasyClassify*

- The method `EClassifier::GetHeatMap` now returns an `EImageC24` instead of an `EImageBW8`.

### *EasySegment Unsupervised*

- There can be small differences in the results compared to previous versions (score, segmented regions...).

### *Easy3D*

- You must call the method `E3DViewer::InitRendering` after the creation of a new `E3DViewer` and before the first call to `E3DViewer::Show`.
- The color ramp configuration is now exposed in `E3DViewer` with the methods `ShowColorRampLegend`, `HideColorRampLegend`, `SetColorRampGraduationColor`, `GetColorRampGraduationColor`, `SetColorRampLocation`, `GetColorRampLocation`, `SetEnableSmartColorRamp` and `GetEnableSmartColorRamp`.
  - The `EColorRamp` class and the related methods are not available anymore.

### *EasyGauge*

- For method `EWorldShape::SetSensor`, the parameter `distortionStrength2` has been removed, with no impact on the calibration process and result.
- The method `EWorldShape::SetDistortion` has been deprecated in favor of `EWorldShape::SetDistortionStrength`. The parameter `distortionStrength2` has been removed, with no impact on the calibration process and result.
- The method `EWorldShape::GetDistortionStrength2` has been removed.

### *Easy*

- Since the addition of the **OEM Key** functionality to the 6514 Neo USB Dongles in addition to the 6512 USB Dongle, the methods `SetOemKey` and `CheckOemKey` require to specify which type of dongle you are using.

## 3.3. Changes

Starting with this release 2.16, Open eVision implements the following changes:

### Easy3D

- *(Breaking change)* You must call the method `E3DViewer::InitRendering` after the creation of a new `E3DViewer` and before the first call to `E3DViewer::Show`.
- Use the method `E3DViewer::SetFontPath` to change the TrueType font used by the `E3DViewer`.
- `EMesh::SaveSTL`, `EPointCloud::SavePCD` and `EPointCloud::SavePLY` now use the binary file format by default instead of the ASCII file format.
- *(Breaking change)* The color ramp configuration is now exposed in `E3DViewer` with the methods `ShowColorRampLegend`, `HideColorRampLegend`, `SetColorRampGraduationColor`, `GetColorRampGraduationColor`, `SetColorRampLocation`, `GetColorRampLocation`, `SetEnableSmartColorRamp` and `GetEnableSmartColorRamp`.
  - The `EColorRamp` class and the related methods are not available anymore.
- When `smartColorGeneration` is enabled, the color ramp in `E3DViewer` now shows explicitly the smallest and the biggest values (first and last tick) outside the linear color ramp between the inliers (second to previous-to-last tick).
- The methods `EPointCloud::AddPointAndAttributesTo` and `EPointCloud::AddPointCloud` have an additional boolean parameter that specifies whether to add attributes present in the source cloud but not in the destination cloud. This parameter is set to `TRUE` by default while previous compoment was equivalent to `FALSE`.

### EasyGauge

- *(Breaking change)* For method `EWorldShape::SetSensor`, the parameter `distortionStrength2` has been removed, with no impact on the calibration process and result.
- *(Breaking change)* The method `EWorldShape::SetDistortion` has been deprecated in favor of `EWorldShape::SetDistortionStrength`. The parameter `distortionStrength2` has been removed, with no impact on the calibration process and result.
- *(Breaking change)* The method `EWorldShape::GetDistortionStrength2` has been removed.

### EasySegment Unsupervised

- The default value for the sampling density parameter is now 1.25 instead of 2. This makes training and inference faster for large images with a limited reduction in detection performances.

### EasyQRCode

- In the `QrcReading` sample, you can now select the Micro QR Code as search model.

### Open eVision Studio

- The **EasyQRCode** tool supports Micro QR Code.
- You can serialize an `EasyQRCodeReader` object with the **EasyQRCode** tool.

### Easy

- (*Breaking change*) Since the addition of the **OEM Key** functionality to the 6514 Neo USB Dongles in addition to the 6512 USB Dongle, the methods `SetOemKey` and `CheckOemKey` require to specify which type of dongle you are using.

### EasyMatrixCode2

- The `MxcLearnAndRead` sample now uses **EasyMatrixCode2** and allows to read multiple codes in an image.

## 3.4. Solved Issues

The following issues have been fixed in Open eVision 2.16:

### Easy3D

- (2.16.0) The method `SaveMetadata` of `EZMap` now saves the world - map transforms and they are not lost when saving and loading metadata anymore.
- (2.16.0) In **Easy3DGrab**, the pan / zoom has been fixed in the demo application ZMap window.
- (2.16.0) The mesh is now cleared before loading an STL file.
- (2.16.1) Saving an empty point cloud to PLY or PCD file does not crash anymore.
- (2.16.1) The adaptive display decimation of `E3DViewer` is now working properly.
- (2.16.1) A bug occurring when converting a mesh or a point cloud to a ZMap has been corrected.
  - The ZMap size was not rounded correctly, thus, when no extension was enabled (`SetExtension` method), some points/faces were missing from the projected cloud/mesh.
  - A side effect of this correction is that, in most cases, the ZMap size is 1 pixel larger than previously. When neither ZMap size nor ZMap resolution are set, in most cases, the resolution is now also a bit smaller.

### EasyMatch

- (2.16.0) When `AdvanceLearning` is set to `TRUE` (default mode), the learning time could be very long. This has been fixed.
- (2.16.0) The `Get` functions are now defined as `const`.

### EasyGauge

- (2.16.1) A display inconsistency when drawing the selected path of an `ERectangleGauge` has been corrected.

### EasyQRCode

- (2.16.1) In some cases, the decoding failed because the micro QR code is positioned too close to the image border. This has been fixed.

### EasyMatrixCode1

- (2.16.1) The same image read in different contexts could return different results . This has been fixed.



## EasyObject2

- (2.16.0) The method `ECodedImage2::DrawObject`, that takes a color as input, is now working correctly again.
- (2.16.1) A crash could happen when giving to **EasyObject2** an ROI where some of the pixels are outside of the images. Now, **EasyObject2** throws an exception in such a case.

## EasyDeepLearning

- (2.16.0) The loading of some old exported datasets has been fixed.
- (2.16.0) Various bugs and crashes when using multiple GPUs have been fixed.
- **EasySegment Supervised and Unsupervised**
  - (2.16.0) A bug when using the scale parameter that could make a training crash has been fixed.
  - (2.16.1) A problem that could generate wrong results and training when setting more than 1 CPU processing thread with `Easy::SetMaxNumberOfProcessingThreads` has been fixed.
- **EasySegment Supervised**
  - (2.16.0) The training required that the images were also labeled for classification. Now, all images with a ground truth segmentation are used during the training.
- **EasyClassify**
  - (2.16.0) The label weights are now correctly taken into account during the training.

## Deep Learning Studio

- (2.16.0) The manipulation of the region of interest has been fixed and does not result in unintended modifications to the region of interest anymore.

## Open eVision Studio

- (2.16.0) The code generation for `EOCR2` recognition has been fixed and is now correct.
- (2.16.0) The loading of a new image from any tab in **EasyMatrixCode** now resets the displayed results from the previous image.

## Linux

- (2.16.1) For parallel processing, the maximum number of processing threads is now properly taken into account.

# 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:

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

becomes:

```
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.  
See: <https://docs.microsoft.com/en-us/visualstudio/debugger/finding-memory-leaks-using-the-crt-library>
  - 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#

```
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#

```
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++

```
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#

```
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".
  - Use UTF-8 encoded strings to handle filenames with non-latin characters.
- `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 functionality 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.

## Easy3D

---

- (2.16.0) When using the class `EMeshToZMapConverter` without extension, some triangles of the mesh close to the ZMap border may be removed.

## EasyDeepLearning

---

- (2.15.0) The Deep Learning tool objects (`EClassifier`, `EUnsupervisedSegmenter`, `ESupervisedSegmenter` and `ELocator`) can leak CPU and GPU memory at destruction. So, it is not recommended to create and delete a lot of Deep Learning tools in the same program.

## 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.