



Release Notes mViz 4.2

This release fixes a number of minor issues. Powerful line and circle fitting functions have been added, as well as assessment of linearity or circularity. They operate on Path data. Some blur and edge detection filters have been added.

A new utility program, mViz Code2D, has been added to ease the setting of the dot code reader for difficult cases and optimize the parameters.

General

We recall that all angles, either user-input or computed by mViz are expressed in Degrees by default. You can change this behavior by setting

```
Status::AngleUnit= Status::FromRadians;  
Status::UnitAngle= Status::ToRadians;
```

Both assignments are required.

The Image class now provides a method Area which returns the number of pixels in the current Window or whole image.

The Status::License function has been improved for easier assessment of plain/temporary license and presence/absence of a dongle. Note that for security reasons, the full documentation of this function is only delivered upon request to the Technical Support.

Image processing

Several new fast filters have been added to support Gaussian and Canny filtering. They appear as Kernel::Binomial/5/7/9 lowpass filters, and as Canny/1/15/2 for the GradientType of the Kernel::Canny and EdgeMap::GradientVector/GradientMaxima/StrongEdges methods.

A filter Kernel::CannyPhase has been added, supporting one the above Canny GradientType. It returns the orientation angle of the gradient vector as a value in range $[0..255]$, corresponding to the angular ranges $[0..360]$ or $[0..180]$ (polarity invariant).

Blob Analysis

The computation of features that require the gray values from the source image, namely Mass, Centroid, Ellipsoid, Average, Deviation, Minimum, Maximum and Histogram required the Window of the source image to be reset, for proper computation. This is no more required.

Code Reading

The Extended Code 39 barcodes could fail to be properly decoded in case they included a non-terminating asterisk * character. This has been fixed.

Gauging

The Path object now supports fitting of a line or circle model to the points. This can be made in a standard or robust way, i.e. when there are alien “garbage” points in the data. Check the methods Path::FitLine and Path::FitCircle. Notice that these advanced functions require the Gauging or Calibration license.

The methods Path::LineDeviation and Path::CircleDeviation (for linearity and circularity assessment) have been enhanced to support explicit specification of the reference line/circle position by the user. The latter can also be obtained automatically by the above FitLine/FitCircle methods. This change will impact the code using them.

The negative indexing (i.e. n^{th} from end) was not working correctly in the Decode method of the Edge gauges when a single edge polarity (Raising or Falling) was chosen. This has been fixed.

Template matching

The Locator object now exposes the method EdgePoints, which retrieves the outline of a given instance of the template and stores them in a Path object. This method is meant for display purposes.

Release Notes mViz 4.1

This is an intermediate release with miscellaneous improvements.

General

The Limits objects (rectangle/window limits) can now save their settings and retrieve them from a file. Check the methods Read and Write.

The Quad object now has a SetCenter method with less arguments, to create a circle with no worries on the arguments.

Blob Analysis

The Blobs context now has a method to pick the blob with the smallest or largest feature value. Check the method Blobs::ByFeatureFirst.

The Centroid and Ellipsoid features of a completely black blob (all pixel values equal zero) cannot be computed. This was causing an arithmetic condition. It is now handled correctly and the corresponding features are left as not-computed.

Image processing

The method Histogram::ShowNoise has been added to the image quality functions. It gives visual feedback about the intensity and distribution of noise in an image.

Methods `Path::LineDeviation` and `Path::CircleDeviation` have been added to assess straightness (smoothness) of straight and curves edges found in a path.

New Gradient types have been added. They correspond to Canny filters of different strengths. Check the `GradientTypes` enumeration.

The `Morpho::Watershed` image segmentation method now accepts images of type vector gradient (rather than grayscale gradient), such as those computed by the class `EdgeMap`.

Calibration

The fitting of a `Scaling` model could result in wrong calibration parameters. This has been fixed.

Code Reading

The 2D code reader now supports an “inking correction” feature. When the cells are too thin or too fat, morphological processing can be applied internally to improve the decoding rate. Check the property `InkingCorrection`. It should be clear that this feature is also usable with direct part marking.

The detection of the QR codes has been enhanced. Some cases such that one of the finder pattern (corner) is damaged can now be decoded.

The Grid grading Quality Indicators were not properly computed in mViz 4.0. This has been fixed.

Gauging

The `EdgeRectangle` widget has a new working mode to measure thicknesses. The new property `FourSides` is true by default, which fits a rectangle. But when set to false, it will only fit two parallel sides, so that their distance can be obtained.

The `EdgeArc` object was not saving all required position properties with the methods `Read/Write`.

The functions `EdgeRectangle::HitHandle` and `EdgeRectangle::Drag` were not handling correctly the `LengthHandle` (sizing of the space around the rectangle edges). This has been fixed.

mVizNET

The Value objects `Site` and `XY` were missing many methods available in the native libraries. They have been restored.

mViz+

Some floating-point parameters such as `CharReader::GapFraction` or `Geometry::CalibrateTarget::OriginRow/Column` could not be set because of bad behavior of the widget. This has been fixed.

The method `Geometry::Undistort` following `Geometry::CalibrateTarget` was not operating. This has been fixed.

The method `Blobs::ByFeatureFirst` has been integrated.

The copy of a constant to a profile via `Process > Point to Point > Profile Arithmetic...` was not possible. This has been fixed.

Some operations on profiles could cause a fatal error because of wrong memory management. This has been fixed.

Note that problems that appear in mViz+ do not necessarily reflect problems in the library mViz.

Release Notes mViz 4.0

The major features of this version of mViz is that the installer becomes 64 bits by default. And more Microsoft compilers are supported: Visual Studio 2008, 2010, 2012, 2013, 2015, and 2017.

The .NET wrapper is now compiled for the framework versions 3.5, 4.0 and 4.6.

The image reader/writer libraries have been upgraded to the most current versions: Png 1.6.37, Jpeg 9c and Tiff 4.0.10.

General

The settings of several mViz objects can now be saved to/retrieved from a disk file (all code readers, all measurement gauges).

The following file extensions are now accepted as aliases when saving/loading images: jpg/jpeg, tif/tiff.

The working range of the `Image::HitHandle` function was too large on a zoomed image. This has been fixed.

Blob Analysis

The Gray1 image type (binary) is now supported for segmentation. Use the overload

```
int Blobs::Segment(const Image& Src, bool Above, bool Connexity8, int MinimumArea, int MaximumArea)
```

The method `Blobs::Segment` working with adaptive thresholding (`int Size` argument) and a mask was not using the mask and applied to the whole image. This has been fixed.

```
void Blobs::Segment(int Size, int Noise, const Region& Mask, const Image& Src, bool Above, bool Connexity8, int MinimumArea, int MaximumArea);
```

Due to a typo, sorting decreasingly on an integer feature was malfunctioning. This has been fixed.

Image processing

Several statistical functions were limited to an image size of $2^{23} = 8\,388\,608$ pixels. This limit has been increased to $2^{31} = 2\,147\,483\,648$. The row width is limited to $2^{15} - 1 = 32767$ pixels.

The method `Histogram::Normalize` has been added. It allows to transform an image by a gain/offset transformation so that its gray mean and standard deviation take specified values. This is useful to deal with images of wildly varying intensity or contrast.

Code Reading

The barcode and 2D reader objects (`Code1D/Code2D Reader`) can now save their settings and retrieve them from a file.

The `Code2DReader` was not appending a null byte at the end of the decoded string when `SymbologyIdentitifer` was activated (though the string length was correct). This has been changed.

Character Reading

The flag `VariableWidth` was not saved correctly to the font files, and was always treated as `true` upon loading. This has been fixed.

The segmentation results of the `CharsSegment` and `CharsRead` methods were slightly different in the `Dotted` modes. This has been fixed.

Gauging

All gauge objects (Edge Point/Line/Arc/Rectangle) can now save their settings and retrieve them from a file. Check the methods Read and Write.

Classification

To avoid name clashes, the data access members Classifier::Bool/Gray/Gray16/Rgb/Int/-Float/Dble had to be renamed with a suffix Value and are now Classifier::BoolValue/-GrayValue/Gray16Value/RgbValue/IntValue/FloatValue/DbleValue.

mVizNET

The Buffer property of an image was not exposed. This has been changed.

Conversions between .NET and mViz Images, and conversely, are now available for all supported types. Note that .NET supports the 16 bits grayscale images very poorly, unlike mViz.

The ref class objects now have a destructor that deallocates unmanaged memory. Use them to force memory deallocation and avoid heap overflow when you create/destroy such objects frequently.

mViz+

The Filter parameter of Geometry::DownSample was set to false by default. This was not the intended behavior, it has been changed.

The operations reading from/writing to a file, now available for several objects, are now accessed via their respective Storage... menu entries.

mViz OCR

Many new font files are available. Check the Images\Fonts folder.