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.