

Vision Library gives you instant access to highest quality, well optimized and field-tested algorithm codes that you may need for your machine vision projects

For processing, analysis, guidance and identification

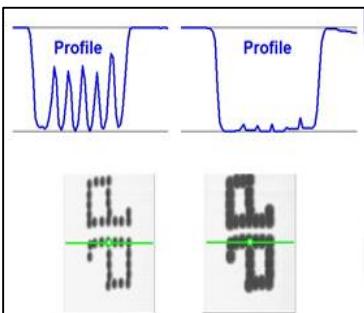
Compatible with the Microsoft C++, C# and Visual Basic environments Linux , 32 and 64 bits

TVC Vision Library offers a comprehensive library of tools that serve as the foundation for building up your machine vision applications.

It is designed for speed, ease of use, and precision that offers OEMs and systems integrators a comprehensive library of vision development tools with high sub-pixel accuracy

TVC is licensed by separate modules or in bundles, run-times per machine, with an attractive pricing scheme. Feature enhancements, porting and special adaptations are possible on request

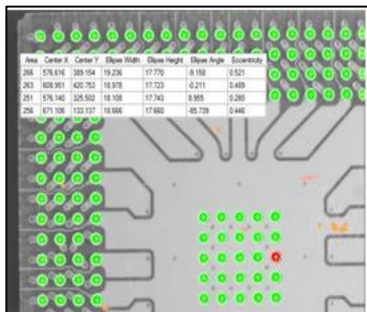
Image Processing



Arithmetic & logic, grayscale & color transforms, shading correction, grayscale morphology, geometric transforms, polar unwarping

```
// Sample code
// Rotate 30°, scale +20% around the center, with interpolation
Geometry::Rotate(XY(320, 240), XY(320, 240), 30, 1.2, Source,
Rotated, true);
// Two points shading correction with Black and White
reference
Operator::Correct(Black, White, Source, Corrected, 0, 255);
```

Image Analysis

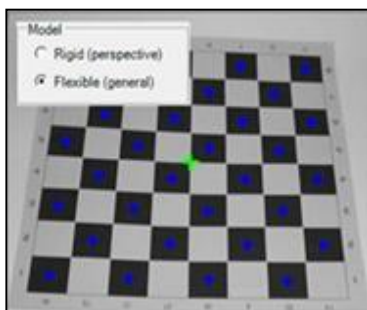


Profile processing, histogram statistics, blob analysis, region masking, contouring

```
// Get the trimmed gray-level mean with 10% trimming double
Mean= Histogram::TrimmedMean(0.1, Source);

// Accumulate the histogram of the first (largest) blob
Blobs.PixelHistogram(0, Source, Histo);
```

Image Calibration



Scaling, isometry, Affinity, Perspective & Distortion Transforms

```
// Adjust the perspective from source to calibrated
points Model.Append(XY(RawX, RawY), XY(GridX,
GridY)); Model.Fit(Perspective);

// Undistort the source image, without bilinear
interpolation Model.Register(Source, Calibrated, false);
```

Edge Gauging

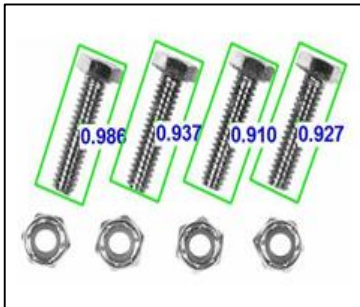


Automatic edge detection, outlier rejection, point model fitting, robust sub-pixel measurement

```
// Place a straight line measurement gauge Gauge.Beg= XY(139, 88); Gauge.End= XY(343, 87);
```

```
// Fit the line and get the angle Gauge.Detect(Source, BestStep); double Angle= Gauge.FittedAngle;
```

Pattern Matching



Model training, sub-pixel location, rotation & scaling

```
// Train the pattern from a region of interest Source.Window(41, 170, 219, 194); Template.Train(Source);
```

```
// NGC search in the whole image with a rotation tolerance Template.MinAngle= -30; Template.MaxAngle= 30; Template.Find(Source);
```

Geometric Matching



Model training, sub-pixel location, full rotation & scaling

```
// Train the pattern from a region of interest Source.Window(41, 170, 219, 194); Template.Train(Source);
```

```
// Geometric search in the whole image with a rotation tolerance Template.MinAngle= -30; Template.MaxAngle= 30; Template.Find(Source, true);
```

Character Reading



Font training, de-skewing, auto-segmentation, printed character recognition & verification

```
// Load the pre-recorded font from a file OCR.Read("Fonts\\OCR-B.fnt");
```

```
// Perform the recognition from a region of interest Source.Window(41, 170, 219, 194); OCR.CharsRead(Source); char* String= OCR.AsciiString;
```

Data Matrix Code



Automatic code location, recognition and decoding, scale and rotation invariant, Data Matrix, Aztec Code, QR...

```
// Find the Data Matrix code from the whole image bool Success= DataMatrix.Decode(Source);
```

```
// Read the decided string char* String= DataMatrix.AsciiString;
```

TVC Machine Library Module Licensing Table:

Module	Classes	Standard	Complete
Image Processing	Profile, Operator, Kernel,, Morpho, Geometry	*	*
Image Analysis	Histogram, Path, Blobs	*	*
Edge Gauging	EdgePoint, -Line, -Arc, -Rectangle	*	*
Pattern Matching	Locator (standard)	*	*
Geometric Matching	Locator (edges)	*	*
Image Calibration	Geometry (advance calibration)	*	*
Character Reading	CharReader		*
Barcode Reading	Code1DReader		*
Dot Code Reading	Code2DReader		*
Code Verification	Code2DReader (check quality)		*