

With  
deep learning  
and 3D tools

# Aurora Design Assistant

Intuitive, versatile, and extendable integrated development environment for machine vision applications

# Overview

## Flowchart-based vision software

Aurora Design Assistant<sup>21</sup>, formerly Matrox Design Assistant, is an integrated development environment (IDE) for Microsoft® Windows® where vision applications are created by constructing an intuitive flowchart instead of writing traditional program code. In addition to building a flowchart, the IDE enables users to design a graphical web-based operator interface for the application.

Aurora Design Assistant can operate independent of hardware, allowing users to choose any computer with CoaXPress®, GigE Vision®, or USB3 Vision® cameras and get the processing power needed. Image capture from CoaXPress cameras happens with the use of a [Zebra Rapixo CXP](#) frame grabber. Aurora Design Assistant works with multiple cameras all within the same project, or per project running concurrently and independently from one another, platform permitting. This field-proven software is also a perfect match for a Zebra [vision controller](#) or [smart camera](#). Aurora Design Assistant offers the freedom to choose the ideal platform for any vision project.

### Why a flowchart?

The flowchart is a universally accessible, recognized, and understood method of describing the sequence of operations in a process. Manufacturing engineers and technicians in particular have all been exposed to the intuitive, logical, and visual nature of the flowchart.

## Aurora Design Assistant at a glance

**Solve machine vision applications** efficiently by constructing flowcharts instead of writing program code

**Choose the best platform for the job** within a hardware-independent environment that supports Zebra [smart cameras](#) and [vision controllers](#) and third-party PCs with CoaXPress, GigE Vision, or USB3 Vision cameras

**Tackle machine vision applications with utmost confidence** using field-proven tools for analyzing, classifying, locating, measuring, reading, and verifying

**Leverage deep learning for visual inspection** through image classification and segmentation tools

**Use a single program** for creating both the application logic and operator interface

**Work with multiple cameras** all within the same project or per project running concurrently and independently from one another, platform permitting

**Interface to [Zebra Altiz](#) and third-party 3D sensors** to visualize, process, and analyze depth maps and point clouds

**Rely on a common underlying vision library** for the same results with a Zebra [smart camera](#), [vision system](#), or third-party computer

**Maximize productivity** with instant feedback on image analysis and processing operations

**Receive immediate, pertinent assistance** through an integrated contextual guide

**Communicate actions and results to other automation and enterprise equipment** via discrete Zebra I/Os, RS-232, and Ethernet (TCP/IP, CC-Link IE Field Basic, EtherNet/IP<sup>22</sup>, Modbus®, OPC UA, and PROFINET®, and native robot interfaces)

**Test communication with a programmable logic controller (PLC)** using the built-in PLC interface emulator

**Maintain control and independence** through the ability to create custom flowchart steps

**Increase productivity and reduce development costs** with Vision Academy [online](#) and [on-premises](#) training

**Protect against inappropriate changes** with the Project Change Validator tool

# Overview (cont.)

## Application design

Flowchart and operator interface design are done within the Aurora Design Assistant IDE hosted on a computer running 64-bit Windows. A flowchart is put together using a step-by-step approach, where each step is taken from an existing toolbox and is configured interactively. Inputs for a subsequent step—which can be images, 3D data, or alphanumeric results—are easily linked to the outputs of a previous step. Decision-making is performed using a flow-control step, where the logical expression is described interactively. Results from analysis and processing steps are immediately displayed to permit the quick tuning of parameters. A contextual guide provides assistance for every step in the flowchart. Flowchart legibility is maintained by grouping steps into sub-flowcharts. A recipes facility enables a group of analysis and processing steps to have different configurations for neatly handling variations of objects or features of interest within the same flowchart.

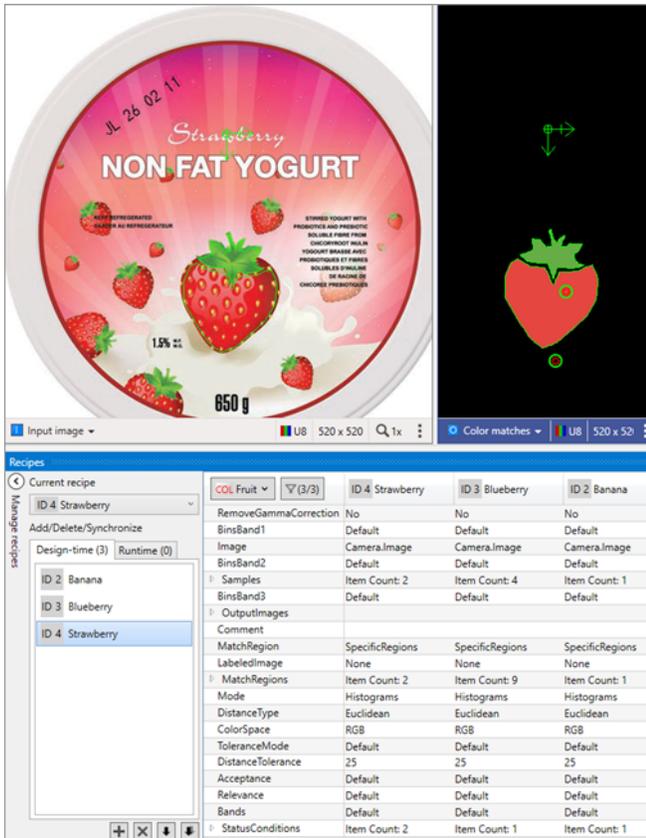
## Latest key additions and enhancements

New deep neural networks for classification and segmentation

Updated [Aurora Imaging Library CoPilot](#) companion application for simplified deep learning training

3D data display in the [Operator View](#)

Support for IEEE 1588 Precision Time Protocol (PTP) time-stamps for GigE Vision acquisition

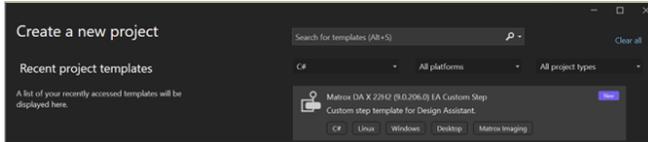


In addition to flowchart design, Aurora Design Assistant enables the creation of a custom, web-based operator interface to the application through an integrated HTML visual editor. Users alter an existing template using a choice of annotations (graphics and text), inputs (edit boxes, control buttons, and image markers), and outputs (original or derived results, status indicators, and charts). A filmstrip view is also available to keep track of and navigate to previously analyzed images. The operator interface can be further customized using a third-party HTML editor.

# Overview (cont.)

## Custom flowchart steps

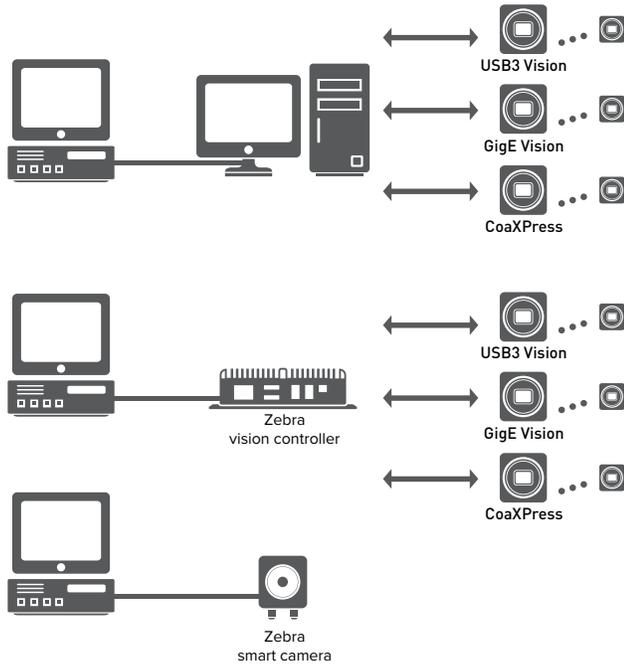
Users have the ability to extend the capabilities of Aurora Design Assistant by way of the included Custom Step software development kit (SDK). The SDK, in combination with Microsoft Visual Studio® 2019 or 2022, enables the creation of custom flowchart steps using the C# programming language. These steps can implement proprietary analysis and processing, as well as proprietary communication protocols. The SDK comes with numerous project samples to accelerate development.



Custom Step SDK

## Application deployment

Once development is complete, the project—with flowchart(s) and operator interface(s)—is deployed either locally or remotely. Local deployment is to the same computer or Zebra vision controller as was used for development. Remote deployment is to a different computer, including Zebra vision controllers, or a Zebra smart camera.



Deployment options

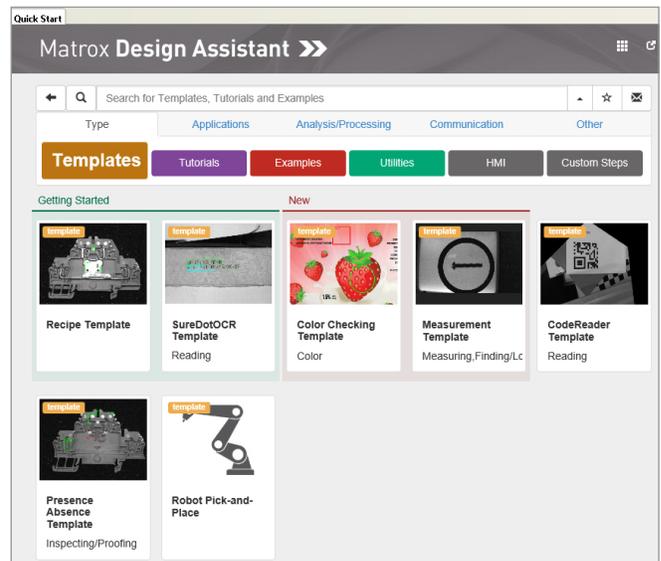
## Project templates for quicker start-up

Aurora Design Assistant includes a series of project templates and video tutorials to help new developers get up and running quickly.

These templates serve as either functional applications or application frameworks intended as a foundation for a target application. Templates also permit dynamic modifications, allowing users to tweak functionality at runtime and immediately see the outcome of any adjustments. The project templates address typical application areas, with examples for:

- Barcode and 2D code reading
- Measurement
- Presences/absence
- Recipes
- Robot guidance (pick-and-place)
- Dot-matrix text reading (SureDotOCR®)
- Color checking

More information on templates can be found on the Quick Start page of Aurora Design Assistant software.



Project templates

# Integrated Development Environment

## Customizable developer interface

The Aurora Design Assistant user interface can be tailored by each developer. The workspace can be rearranged, even across multiple monitors, to suit individual preferences and further enhance productivity.

The screenshot displays the Aurora Design Assistant interface with several key components:

- Flowchart:** A vertical flowchart on the left shows steps: Loop, CAM Camera, CIRC CircleFinder, VOL3D Volume3D, INTG CheckIfBroken, STA Status, and P.
- 3D Data:** Two 3D visualizations of a point cloud are shown. The left one is a top-down view, and the right one is a perspective view with a wireframe overlay. Camera settings (Azimuth: 56°, Elevation: 26°, Roll: 0°) are visible.
- Annotations:** A panel on the right lists 'VOL3D Volume3D' with sub-items 'Reference' and 'Regions'. A 'Crop regions' checkbox is checked, and a 'Draw...' button is present.
- Quick Access:** A panel on the far right provides context for the 'Crop region' operation, explaining that the surface is composed of points from a point cloud or depth map.
- Step Log:** A log on the right shows the execution sequence: FlowchartStart, Loop (20), Camera (Image4.mbufc), CircleFinder, and Volume3D.
- Configuration Dialog:** A 'Volume3D' configuration window is open, showing 'Enabled crop region' checked and 'Shape: Cylinder3D' selected. It includes fields for 'Relative to' (FromFixture2D) and 'Fixture 2D' (CircleFinder.Occurrences(1).Fixture).
- Results Table:** A table at the bottom right shows execution results for 'WarningGaps'.

Volume status	Volume	Execution Time
WarningGaps	26153.7	0.0635282 s
WarningGaps	26223.5	0.0560752 s
WarningGaps	18713.5	0.0457051 s
WarningGaps	25554.8	0.0547719 s
WarningGaps	26496.7	0.0558995 s
WarningGaps	27208.7	0.0567304 s
WarningGaps	19710	0.0488067 s
WarningGaps	11955.9	0.0368726 s
WarningGaps	26153.7	0.0571482 s
WarningGaps	26223.5	0.0561529 s
WarningGaps	18713.5	0.0474662 s
WarningGaps	25554.8	0.0547719 s

Annotations:

- VOL3D Volume3D
- Reference
- Regions
- Crop regions  Draw...

Quick Access

- Crop region

The Surface of the volume is composed of points from the point cloud or depth map. Crop operations select those points. The crop is done based on one or more 2D region(s), on a 3D shape, on non zero pixels in a mask, or on the position relative to the Z=equals-zero

Properties Quick Access

Step Log

- FlowchartStart
- Loop (20)
- Camera (Image4.mbufc)
- CircleFinder
- Volume3D

Configuration

Volume3D

VolumeOk Volume3D.Volume/1000 26.1537 INRANGE 26 to 28

GENERAL

- Enabled crop region

Region mode:  Positive Z  Shape 3D  Regions  Mask image

Shape: Cylinder3D Center the shape Use existing...

Fixture definition Shape parameters

Relative to: FromFixture2D

Fixture 2D: CircleFinder.Occurrences(1).Fixture Z: 0

Results

Volume status	Volume	Execution Time
WarningGaps	26153.7	0.0635282 s
WarningGaps	26223.5	0.0560752 s
WarningGaps	18713.5	0.0457051 s
WarningGaps	25554.8	0.0547719 s
WarningGaps	26496.7	0.0558995 s
WarningGaps	27208.7	0.0567304 s
WarningGaps	19710	0.0488067 s
WarningGaps	11955.9	0.0368726 s
WarningGaps	26153.7	0.0571482 s
WarningGaps	26223.5	0.0561529 s
WarningGaps	18713.5	0.0474662 s
WarningGaps	25554.8	0.0547719 s

View flowchart with images/3D data or operator interface layout

Get quick access to contextual documentation

Configure each step through a dialog (or directly on image/3D data)

View step execution results in tabular form (or directly on image/3D data)

Track flowchart execution

# Operator View

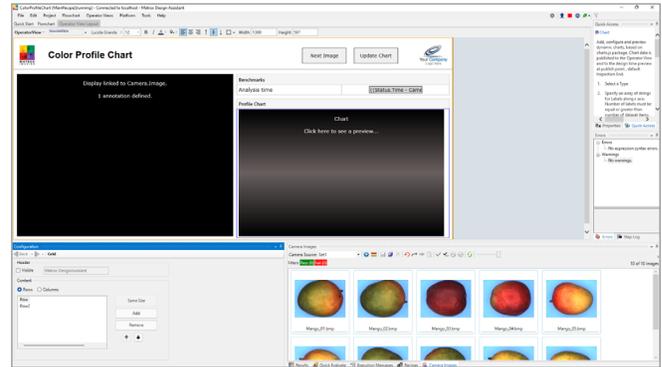
## Operator interface viewable anywhere

The web-based operator interface, or Operator View, can be accessed locally or remotely through a HTML-5-capable web browser such as Google Chrome™ and Chromium, Mozilla Firefox®, and Microsoft Internet Explorer® and new Edge. Local viewing is done on the same computer or Zebra [vision controller](#) as was used for development. Local viewing is also available with a Zebra [smart camera](#) through a simple touch screen connected to its video output and USB interface, which eliminates the need for an additional computer. Remote viewing is done from any computer, including dedicated Human Machine Interface (HMI) or touch-panel PCs.

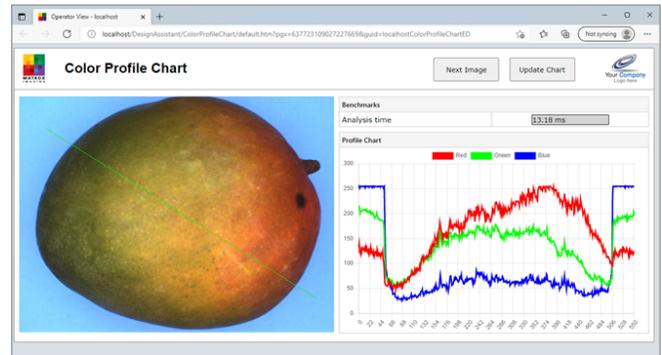
A stand-alone HMI application can be created using Microsoft Visual Studio to run on the local or a remote computer as an alternative to the web-based operator interface.

## Security features

Access to specific Operator Views can be made to require user authentication (i.e., username and password) so only authorized personnel can modify key parameters of a running project. A project can be locked to a specific Zebra [smart camera](#) or [vision controller](#) when deployed, preventing it from running on an unauthorized platform. A project can also be encrypted during deployment to a platform, ensuring that the project cannot be read or changed by unauthorized users. Projects locked to a platform are automatically encrypted.



Design a customized Operator View



Resulting Operator View as seen in a web browser

# Field-Proven Vision Tools

## Image analysis and processing

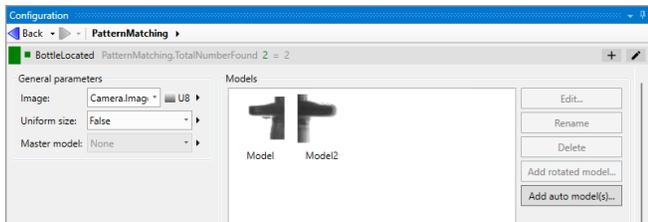
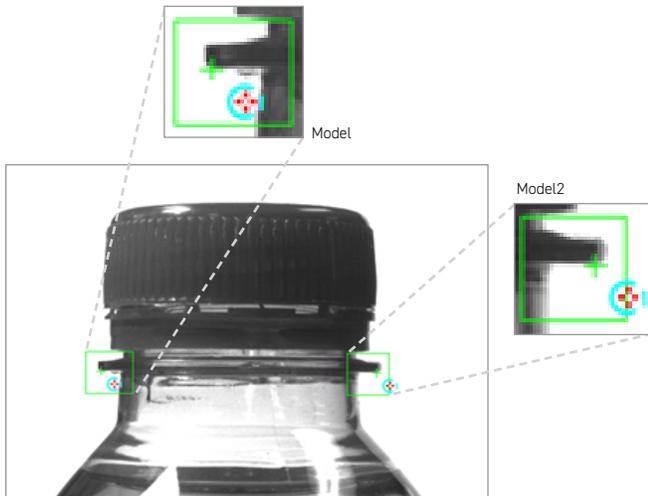
Central to Aurora Design Assistant are flowchart steps for calibrating, enhancing, and transforming images; locating objects; extracting and measuring features; classifying and segmenting images; reading character strings; and decoding and verifying identification marks. These steps are designed to provide optimum performance and reliability.

## Pattern recognition steps

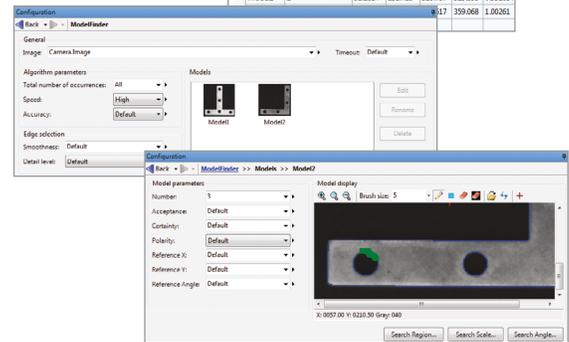
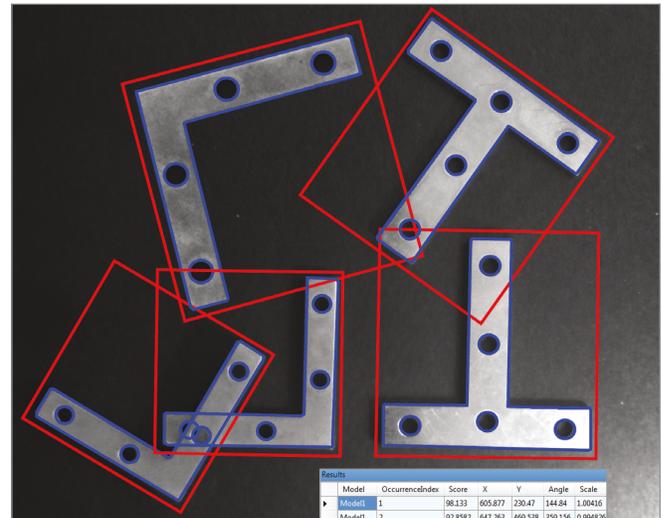
Aurora Design Assistant includes two steps for performing pattern recognition: PatternMatching and ModelFinder. These steps are primarily used to locate complex objects for guiding a gantry, stage, or robot, or for directing subsequent analysis and processing steps.

The PatternMatching step finds a pattern by looking for a similar spatial distribution of intensity. The step employs a smart search strategy to quickly locate multiple patterns, including multiple occurrences, which are translated and slightly rotated. The step performs well when scene lighting changes uniformly, which is useful for dealing with attenuating illumination. A pattern can be trained manually or determined automatically for alignment. Search parameters can be manually adjusted and patterns can be manually edited to tailor performance.

The ModelFinder step employs an advanced technique to locate an object using geometric features (e.g., contours). The step finds multiple models, including multiple occurrences that are translated, rotated, and scaled. Model Finder locates an object that is partially missing and continues to perform when a scene is subject to uneven changes in illumination, thus relaxing lighting requirements. A model is manually trained from an image and search parameters can be manually adjusted and models can be manually edited to tailor performance.



PatternMatching step



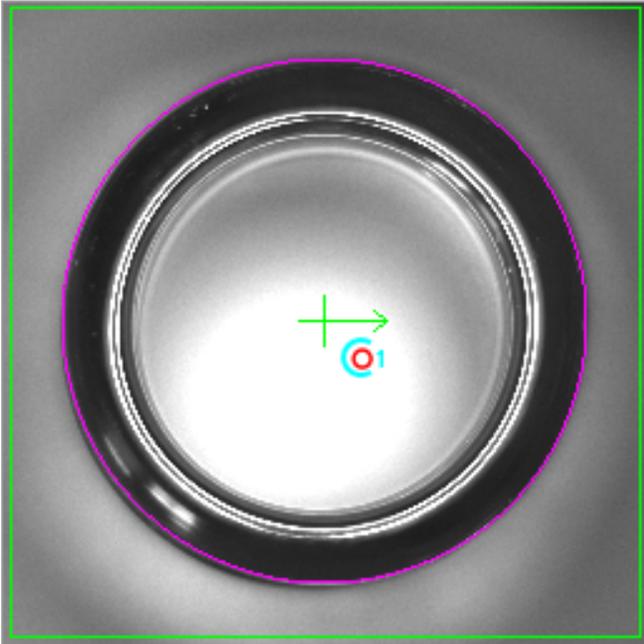
ModelFinder step

# Field-Proven Vision Tools (cont.)

## Shape finding steps

Aurora Design Assistant includes dedicated steps for finding circles, ellipses, rectangles, and line segments. Circle finding is defined by the anticipated radius, the possible scale range, and the number of expected occurrences. Ellipse and rectangle finding are defined by the anticipated width and height, the possible scale and aspect ratio ranges, and the number of expected occurrences. Line segment finding is defined by the anticipated length and the number of expected occurrences. Continuous and broken edges lying within an adjustable variation tolerance produce the requested shape.

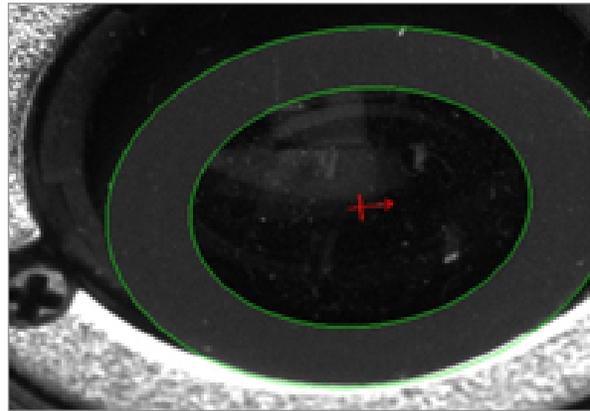
The shape-finding step computes the total number of found occurrences; for each occurrence, the tool can provide the center position and score relative to the reference. It can also give the radius and scale for circles; the angle, aspect ratio, width, and scale for ellipses and rectangles; and the start and end positions as well as the length for line segments. These specialized modes are generally faster and more robust at finding the specific shapes than generic pattern recognition.



CircleFinder step



RectangleFinder step

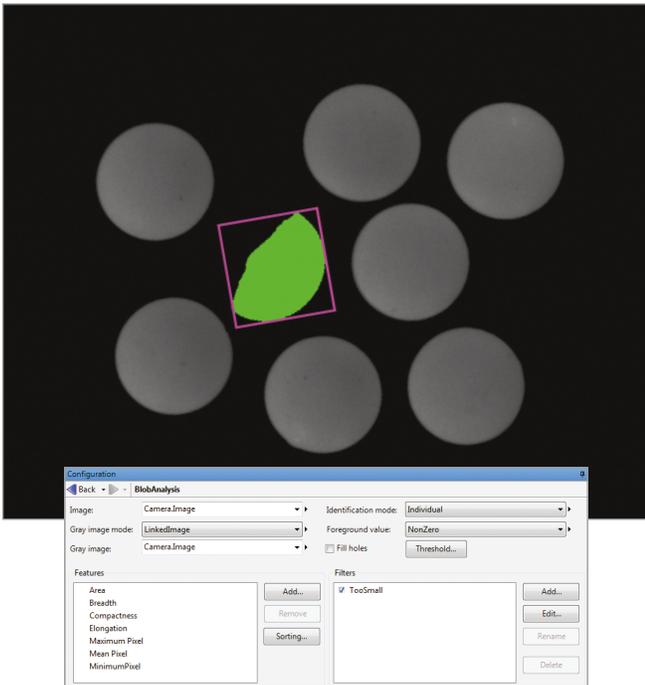


EllipseFinder step

# Field-Proven Vision Tools (cont.)

## Feature extraction and analysis step

The BlobAnalysis step in Aurora Design Assistant is used to identify, count, locate, and measure basic features and objects (i.e., blobs) to determine presence and position, and enable further inspection. The step works by segmenting images where blobs are separated from the background and one another before quickly identifying the blobs. Over 50 characteristics can be measured and these measurements can be used to eliminate or keep certain blobs.



BlobAnalysis step

# Field-Proven Vision Tools (cont.)

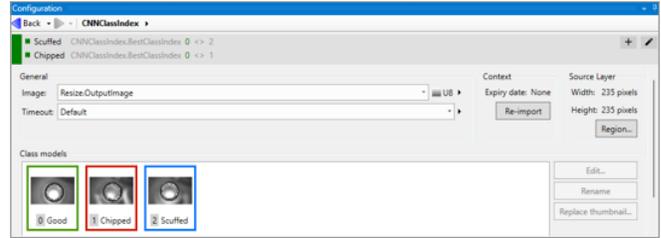
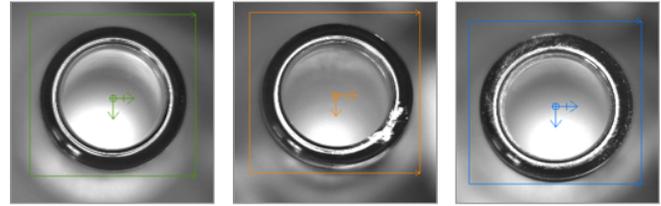
## Classification steps

Aurora Design Assistant includes classification steps for automatically categorizing image content using machine learning. These steps make use of deep learning technology—specifically the convolutional neural network (CNN) and variants—in two distinct approaches.

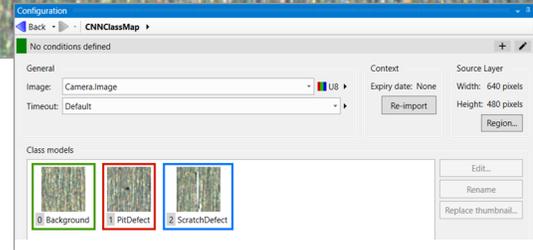
The first or global approach—implemented by the CNNClassIndex step—assigns images or image regions to pre-established classes. It lends itself to identification tasks where the goal is to distinguish between similarly looking objects including those with slight imperfections. The results for each image or image region consist of the most likely class and a score for each class.

The second or segmentation approach—implemented by the CNNClassMap step—generates maps indicating the pre-established class and score for all image pixels. It is appropriate for detection tasks where the objective is to determine the incidence, location, and extent of flaws or features. These features can then be further analyzed and measured using traditional tools like the BlobAnalysis step. These classification steps are particularly well suited for analyzing images of highly textured, naturally varying, and acceptably deformed goods in complex and varying scenes.

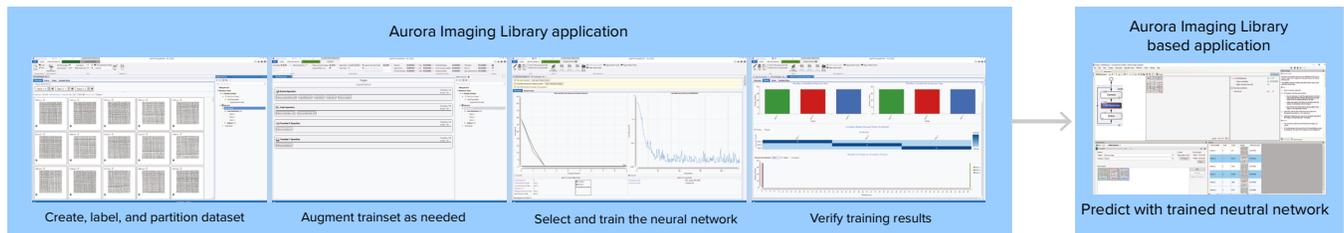
Users can opt to train a deep neural network on their own—using the included Aurora Imaging Library CoPilot application—or commission Zebra to do so using previously collected images that are both adequate in number and representative of the expected application conditions. Different types of training are supported, such as transfer learning and fine-tuning, all starting from one of the supplied pre-defined deep neural network architectures. Aurora Imaging Library CoPilot provides what is needed for building the required training dataset—including the labeling of images and augmenting the dataset with synthesized images—as well as monitoring and analyzing the training process. Training is accomplished using a NVIDIA GPU<sup>3,4</sup> or x64-based CPU while inference is performed on a CPU in a Zebra vision controller, smart camera, or third-party computer, avoiding the need for specialized GPU hardware.



CNNClassIndex step



CNNClassMap step



Aurora Imaging Library CoPilot workflow for Classification training

# Field-Proven Vision Tools (cont.)

## 1D and 2D measurement steps

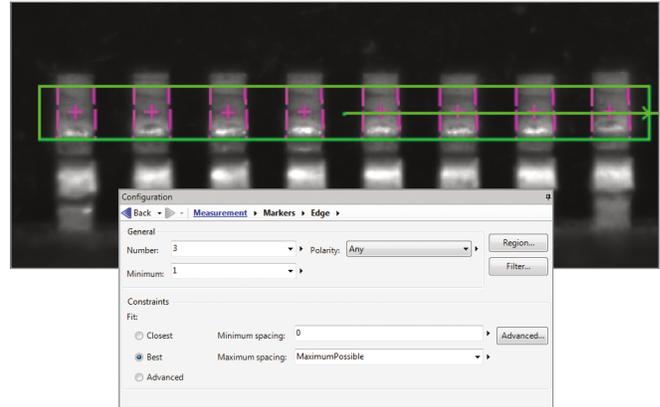
Aurora Design Assistant includes three steps for measuring: Measurement, BeadInspection, and Metrology. These tools are predominantly used to assess manufacturing quality.

The Measurement step uses the projection of image intensity to very quickly locate and measure straight edges or stripes within a carefully defined rectangular region. The tool can make several 1D measurements on edges and stripes, as well as between edges or stripes.

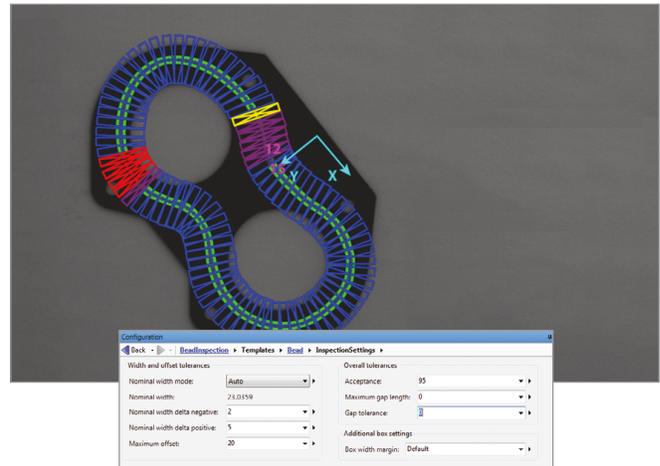
The BeadInspection step is for inspecting material that is applied as a continuous sinuous bead, such as adhesives and sealants, or its retaining channel. The step identifies discrepancies in length, placement, and width, as well as discontinuities. The BeadInspection step works by accepting a user-defined coarse search path as a list of points on a reference bead and then automatically and optimally placing search boxes to form a template. The size and spacing of these search boxes can be modified to change the sampling resolution. The allowable bead width, offset, gap, and overall acceptance measure can be adjusted to meet specific inspection criteria.

The Metrology step is intended for 2D geometric dimensioning and tolerancing applications. The step extracts edges within defined regions to best fit geometric features. It also supports the construction of geometric features derived from measured ones or defined mathematically. Geometric features include arcs, circles, points, and segments. The step validates tolerances based on the dimensions, positions, and shapes of geometric features. The Metrology step can also analyze profiles obtained from 3D profile sensors or extracted from point clouds and depth maps.

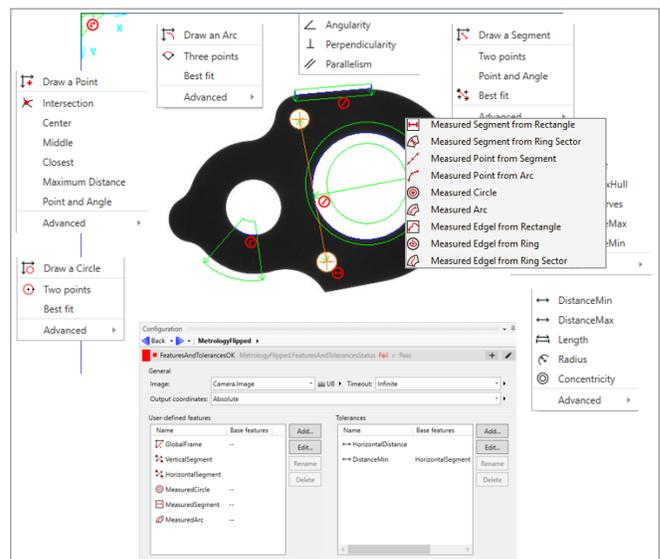
The Metrology step's effectiveness is maintained when subject to uneven changes in scene illumination, which relaxes lighting requirements. The expected measured and constructed geometric features, along with the tolerances, are kept together in a template which is easily repositioned using the results of other locating steps.



Measurement step



BeadInspection step



Metrology step

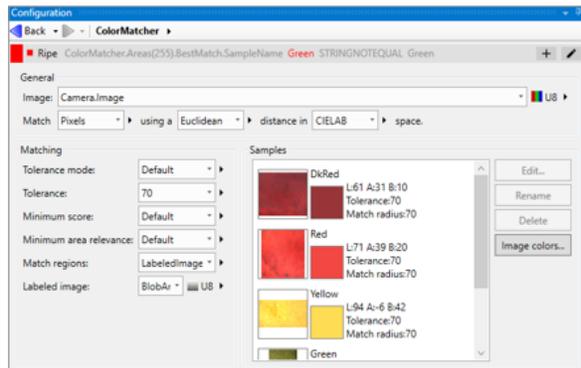
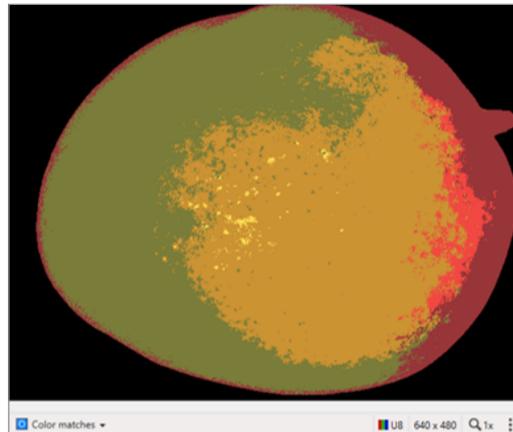
# Field-Proven Vision Tools (cont.)

## Color analysis steps

Aurora Design Assistant includes a set of tools to identify parts, products, and items using color, assess quality from color, and isolate features using color.

The ColorMatcher step determines the best matching color from a collection of samples for each region of interest within an image. A color sample can be specified either interactively from an image—with the ability to mask out undesired colors—or using numerical values. A color sample can be a single color or a distribution of colors (i.e., a histogram). The color matching method and the interpretation of color differences can be manually adjusted to suit particular application requirements. The ColorMatcher step can also match each image pixel to color samples to segment the image into appropriate elements for further analysis using other steps such as [BlobAnalysis](#).

The ImageProcessing step includes operations to calculate the color distance and perform color projection. The distance operation reveals the extent of color differences within and between images, while the projection operation enhances color to grayscale image conversion for analysis using other grayscale processing steps.



ColorMatcher step

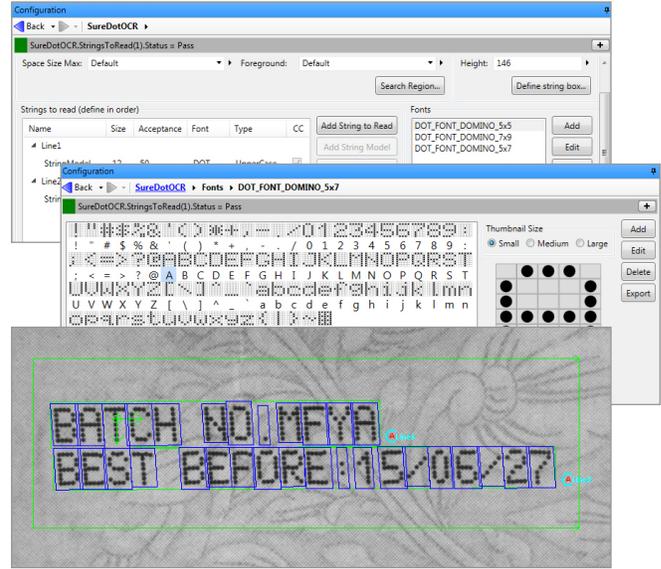
# Field-Proven Vision Tools (cont.)

## Character recognition steps

Aurora Design Assistant provides two steps for character recognition: SureDotOCR and StringReader. These steps combine to read text that is engraved, etched, marked, printed, punched, or stamped on surfaces.

The SureDotOCR step is uniquely designed for the specific challenge of reading dot-matrix text produced by inkjet printers and dot peen markers. Its use is straightforward, just needing to specify the dot size, the number of characters in a text string, and the dimension—but not the location—of the text region. The step reads strings located at any angle, with varying contrast, and on an uneven background. It interprets distorted characters and handles some variability to character scale. The step recognizes punctuation marks and blank spaces. It includes pre-defined fonts that can be edited. The step automatically reads multiple lines of text where each line can be in a different font. It supports user-defined constraints, overall and at specific character positions, to further enhance recognition rates. The SureDotOCR step provides greater robustness and flexibility than case-specific techniques that convert dot-matrix characters into solid ones for reading with traditional character recognition tools.

The StringReader step is based on a sophisticated technique that uses geometric features to quickly locate and read text made up of solid characters in images where these characters are well separated from the background and from one another. It handles text strings with a known or unknown number of evenly or proportionally spaced characters. The step accommodates changes in character angle with respect to the string, aspect ratio, scale, and skew, as well as contrast reversal. It accepts strings located across multiple lines and at a slight angle. The step reads from multiple pre-defined or user-defined Latin-based fonts. It supports user-defined constraints, overall and at specific character positions, to further increase recognition rates.



SureDotOCR step



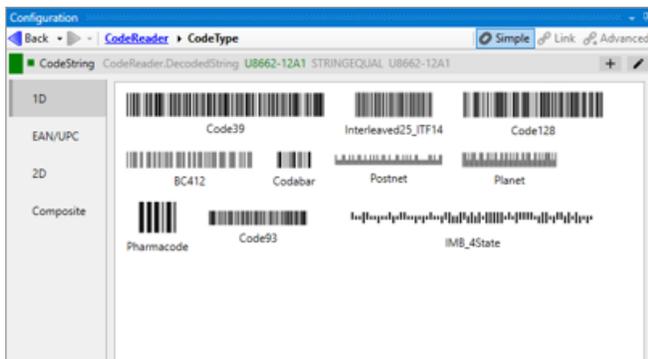
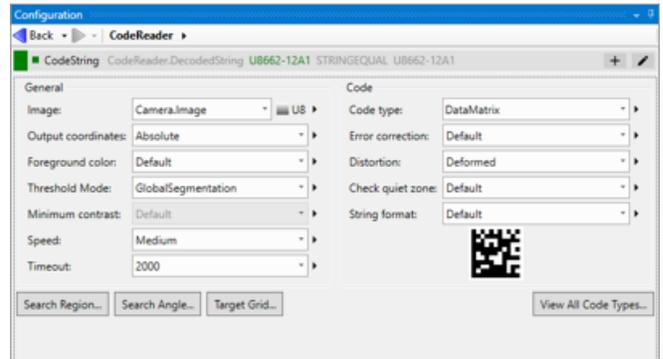
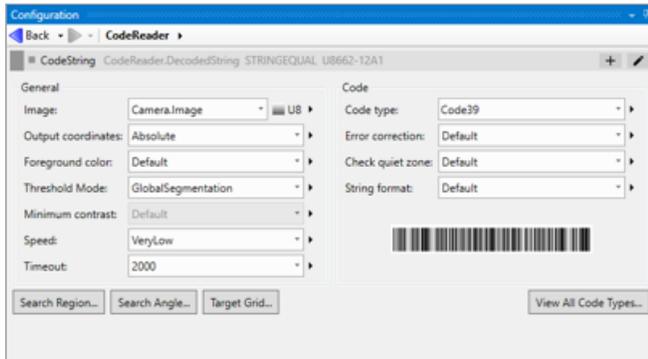
StringReader step

# Field-Proven Vision Tools (cont.)

## 1D and 2D code reading and verification steps

The CodeReader step in Aurora Design Assistant locates and reads 1D, 2D, and composite identification marks. The step handles rotated, scaled, and degraded codes in tough lighting conditions. The step can provide the orientation, position, and size of a code.

In addition, the CodeGrade step verifies the quality of a code based on the ANSI/AIM and ISO/IEC grading standards.



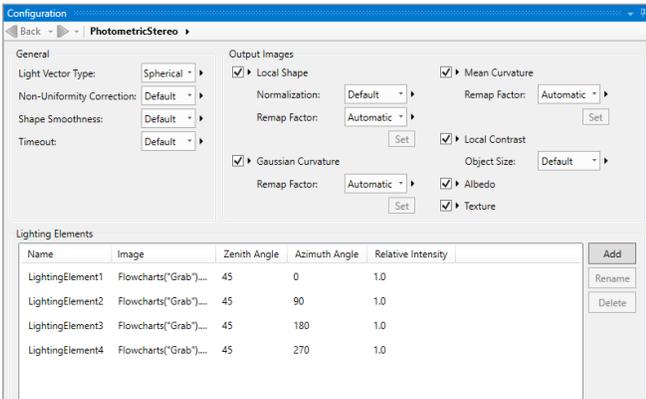
CodeReader step (1D)

CodeReader step (2D)

# Field-Proven Vision Tools (cont.)

## Photometric stereo step

A PhotometricStereo step in Aurora Design Assistant produces an image that emphasizes surface irregularities, such as embossed or engraved features, scratches, or indentations. The image is produced from a series of images of the same scene taken with directional illumination as driven by a Quad (X2) Controller from Advanced Illumination (Ai), a Light Sequence Switch (LSS) from CCS, a LED Light Manager (LLM) from Smart Vision Lights, or similar light controller.

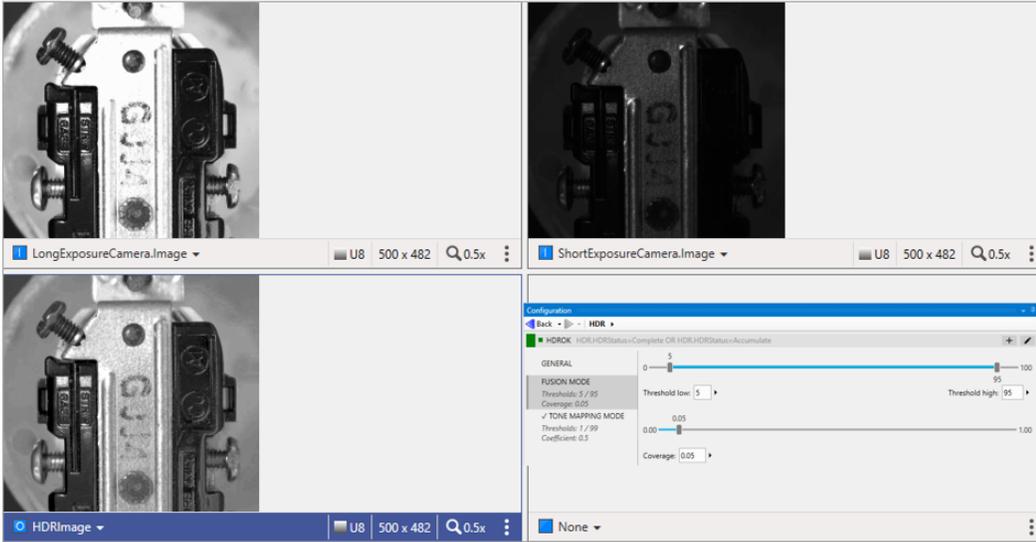


PhotometricStereo step

# Field-Proven Vision Tools (cont.)

## HDR step

An HDR step in Aurora Design Assistant combines images of an identical scene, taken at different camera exposure levels, into a single image that contains a greater range of luminance (contrast).



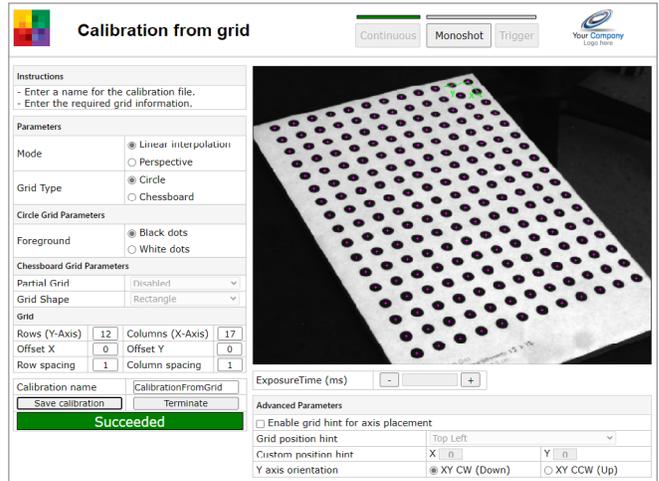
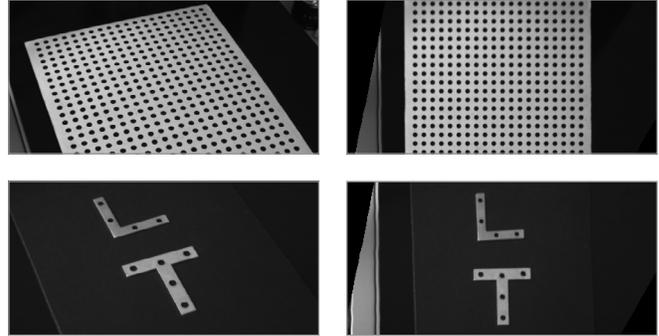
HDR step

# Field-Proven Vision Tools (cont.)

## 2D calibration utility

Calibration is a routine requirement for machine vision. Aurora Design Assistant includes 2D calibration capabilities to convert results (i.e., positions and measurements) from pixel to real-world units and vice-versa. The capability can compensate results and even an image itself for camera lens and perspective distortions.

Calibration is achieved using an image of a grid or just a list of known points and is performed through a utility project accessed from the Aurora Design Assistant management portal.



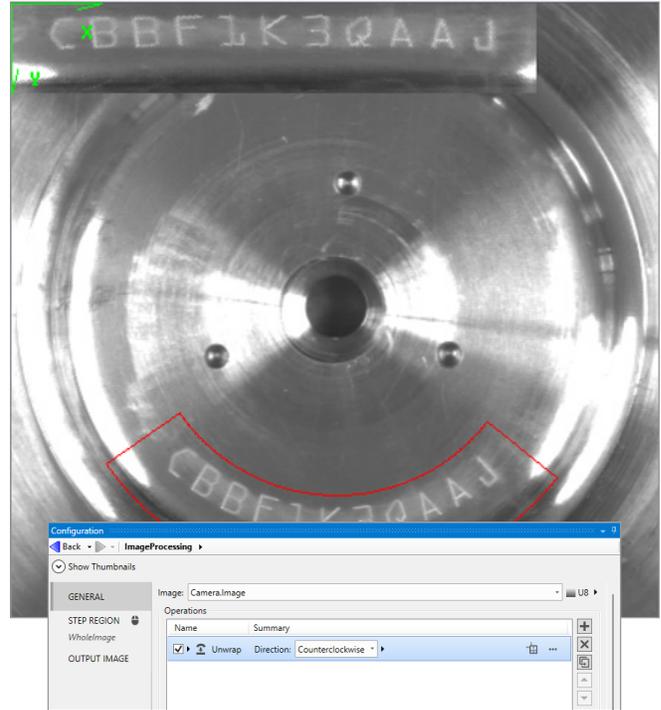
Calibration utility

# Field-Proven Vision Tools (cont.)

## Basic image processing steps

Aurora Design Assistant includes the ImageProcessing step for enhancing and transforming images in preparation for subsequent analysis. Supported operations include arithmetic, color space conversions, color distance and projection (refer to [Color analysis](#) section for details), filtering, geometric transformations, logic, LUT mapping, morphology, and thresholding.

Aurora Design Assistant also includes EdgeLocator and IntensityChecker steps. The EdgeLocator step finds objects by locating straight edges and the IntensityChecker step is used to analyze an object using image intensity.



ImageProcessing step

# Field-Proven Vision Tools (cont.)

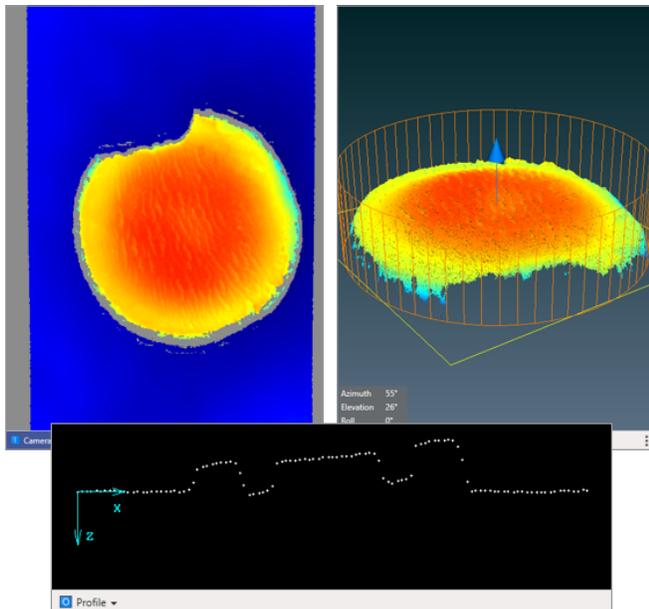
## 3D acquisition and display

Aurora Design Assistant interfaces to a range of 3D cameras and sensors. These include [Zebra Altiz](#) 3D profile sensors, as well as third-party 3D cameras and sensors through their SDK or the interface standard they support. The type of 3D data acquired includes profiles, depth maps, and point clouds. Point-cloud surfaces can be meshed and filled as required by subsequent operations. The software includes an SDK to allow users to create interfaces to 3D camera and sensors on their own.



Zebra Altiz

Profiles and depth maps are visualized as charts and greyscale or color-coded images, respectively. Point clouds and depth maps can be viewed in the 3D display, as is or with a color map which is available both in the IDE or in the Operator View. In the IDE, users also have the option of showing the point clouds as a mesh with or without filling.



Depth map, point cloud, and profile views

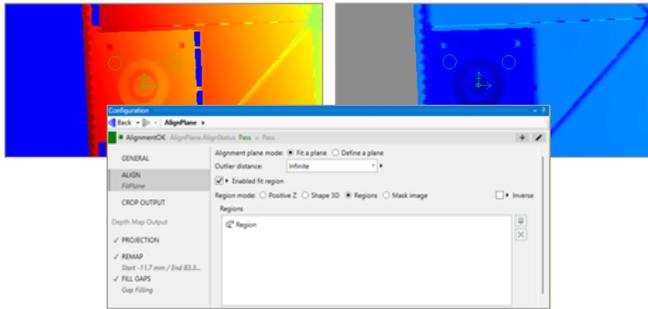
Manufacturer	Model(s)
Automation Technology	C5-CS series
Basler	blaze
FRAMOS	D400e series
Intel	D400 series
LMI	Gocator 2000 and 3000 series
Lucid	Helios(2/2+)
Micro Epsilon	scanCONTROL and surfaceCONTROL
Photoneo	PhoXi series
SmartRay	ECCO series
Wenglor	weCat3D and ShapeDrive series
Zivid	One+ and Two

Supported third-party 3D cameras and sensors. [Contact Zebra](#) for interfaces to other third-party 3D cameras and sensors.

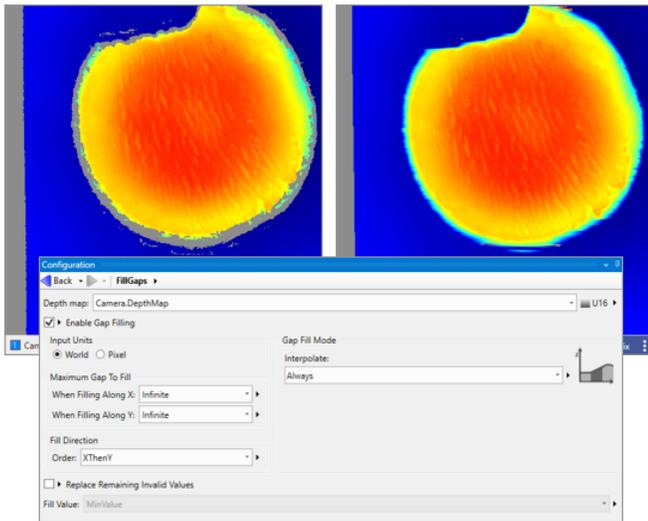
# Field-Proven Vision Tools (cont.)

## 3D analysis and processing

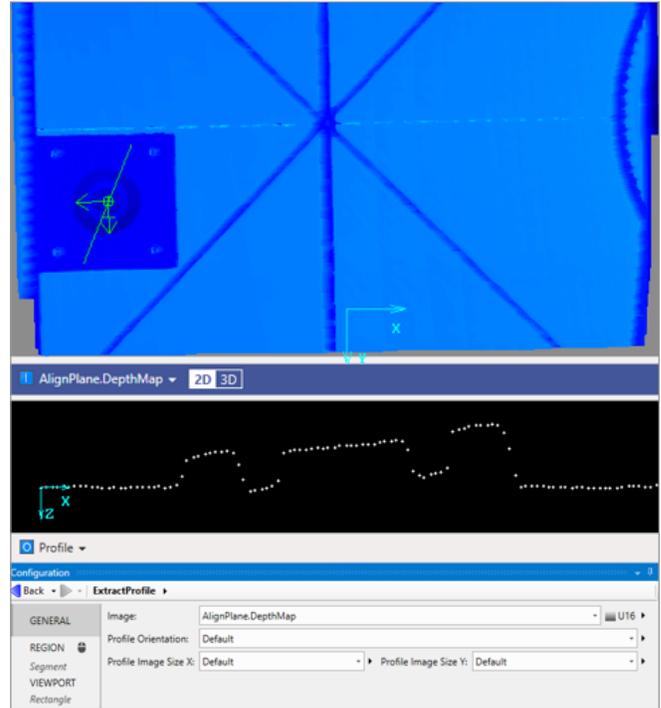
Aurora Design Assistant is able to inspect the data produced by 3D cameras and sensors using the previously mentioned 2D or additional 3D-specific analysis and processing steps. An AlignPlane step is available to re-align the reference or resting surface in a depth map or point cloud to remove disruptive perspective prior to ensuing operations. A Project3D step (as well as an option in the Camera step) is included to obtain a suitable depth map from a point cloud for use with 2D analysis and processing steps. A FillGaps step (as well as an option in the Camera step) is provided to fill in voids and replace invalid data in a depth map. An ExtractProfile step is on hand to extract a profile from a depth map or point cloud for analysis using the Metrology step. A Volume3D step is available to compute the volume of a depth map as well as a point cloud whose surface is meshed and filled. A Crop3D step (as well as an option in the AlignPlane and Volume3D steps) is present to pull out a portion of interest from a point cloud.



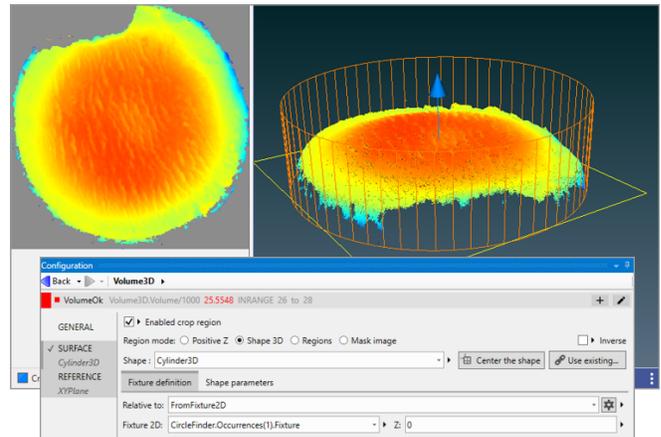
AlignPlane step



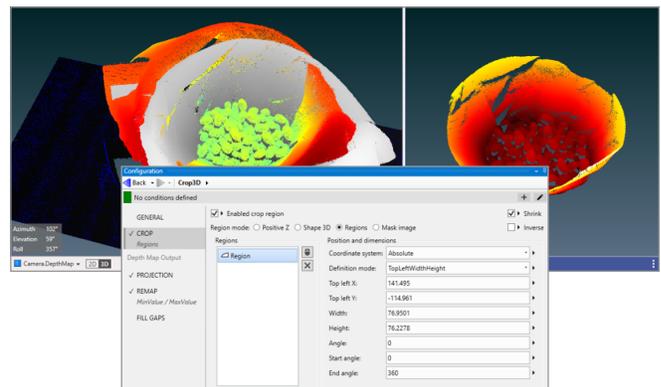
FillGaps step



ExtractProfile step



Volume3D step



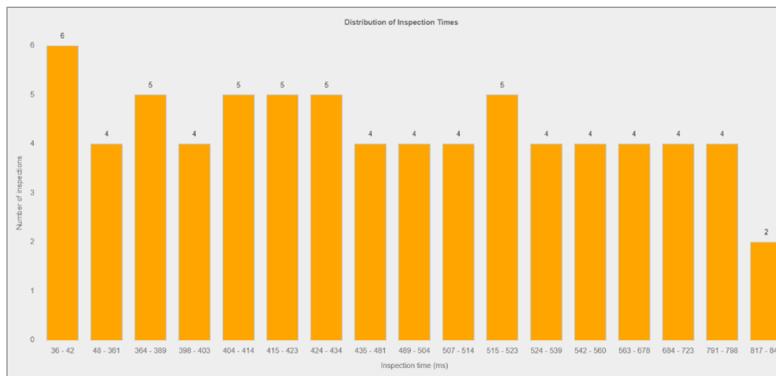
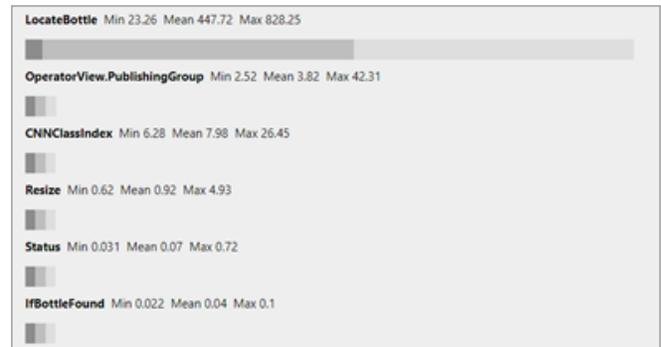
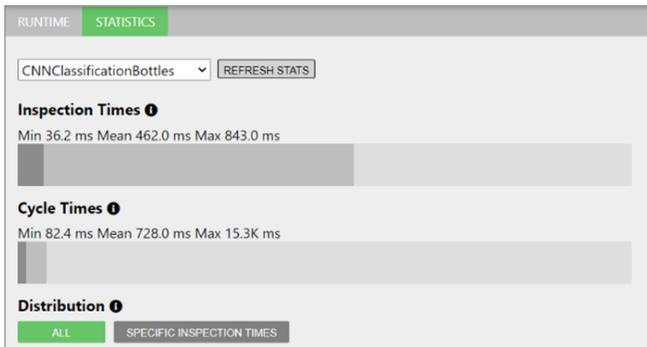
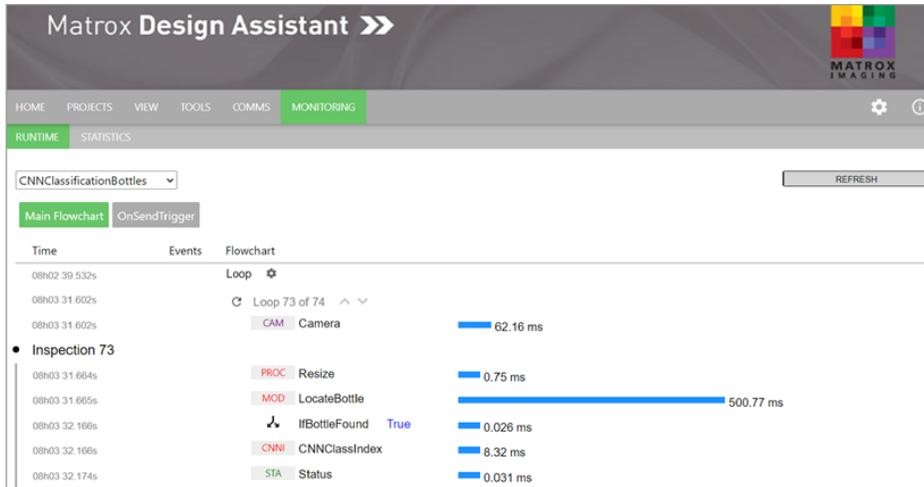
Crop3D step

# Utilities

## Runtime monitor

Aurora Design Assistant’s management portal includes a facility to monitor a deployed and running project and collect performance statistics to aid with troubleshooting and optimization. One of the facility’s pages provides a breakdown of flowchart execution by operation for a given loop. Another page gives the minimum, mean, and maximum execution times overall and by operation as well a

distribution of the total execution times. Aurora Design Assistant also includes Aurora Profiler, a separate utility to post-analyze the execution of a vision project for performance bottlenecks and timing issues.



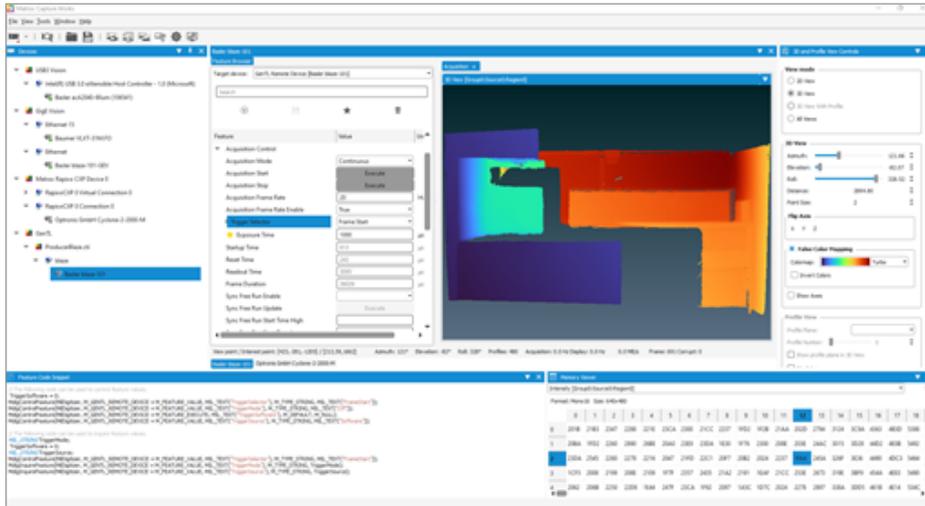
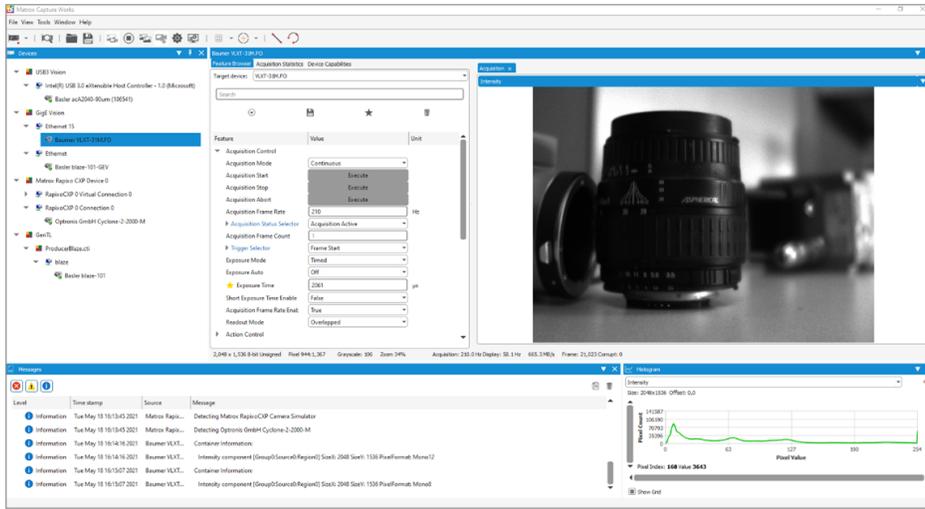
Runtime monitor

# Utilities (cont.)

## Aurora Capture Works

Aurora Design Assistant comes with Aurora Capture Works, a utility for verifying the connection to one or more GenICam™-based cameras or 3D sensors and testing acquisition from these. Aurora Capture Works can obtain CoaXPress, GenTL, GigE Vision, and USB3 Vision device information, collect and present acquisition statistics, and provide access to acquisition properties. The built-in Feature Browser allows the user to configure and control devices

with ease. Device settings can be saved for future reuse. Captured data from multiple devices can be displayed efficiently in 2D and 3D where applicable, with the option to view histograms, 3D and profile data, real-time pixel profiles, memory values, and much more. Aurora Capture Works can also be used to apply firmware updates to devices provided these follow the GenICam FWUpdate standard.



Aurora Capture Works configuration and test tool for GenICam-based interface standards

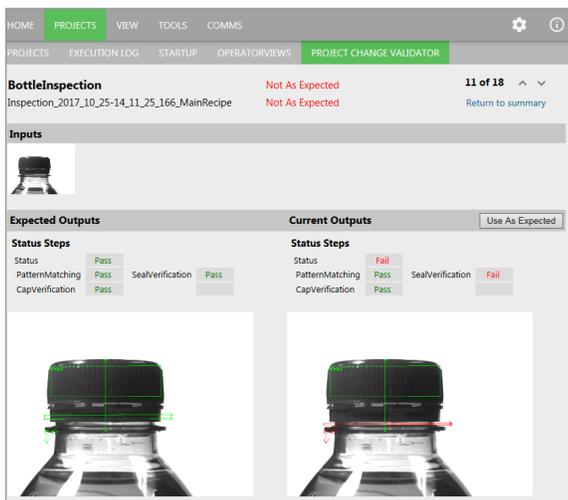
# Utilities (cont.)

## Project Change Validator

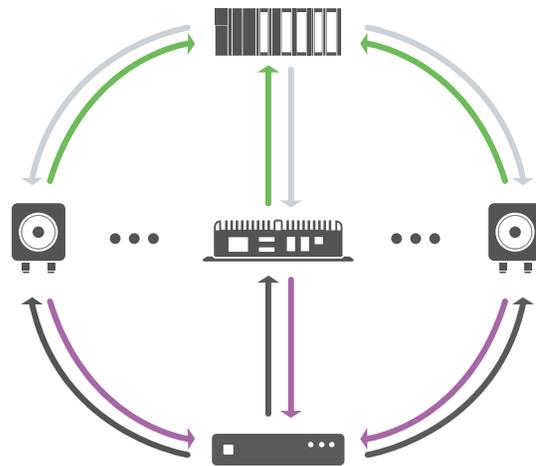
Project Change Validator is a utility employing a client-server architecture for ensuring that changes made to a deployed project are not detrimental to the functioning of that project. It provides the ability to record reference images—along with the associated inspection settings and results—for a given project.

This archived reference data is then used to validate changes made to the project. Changes are validated by running the modified project with the reference data and comparing the projects' operation against this data. Validation is performed by the server—typically running on a separate computer—which is reachable over a network.

The Aurora Design Assistant management portal provides access to the validation data and results. Validation requests are made on demand from the management portal, an automation controller, or an HMI panel.



Project Change Validator (view from management portal)

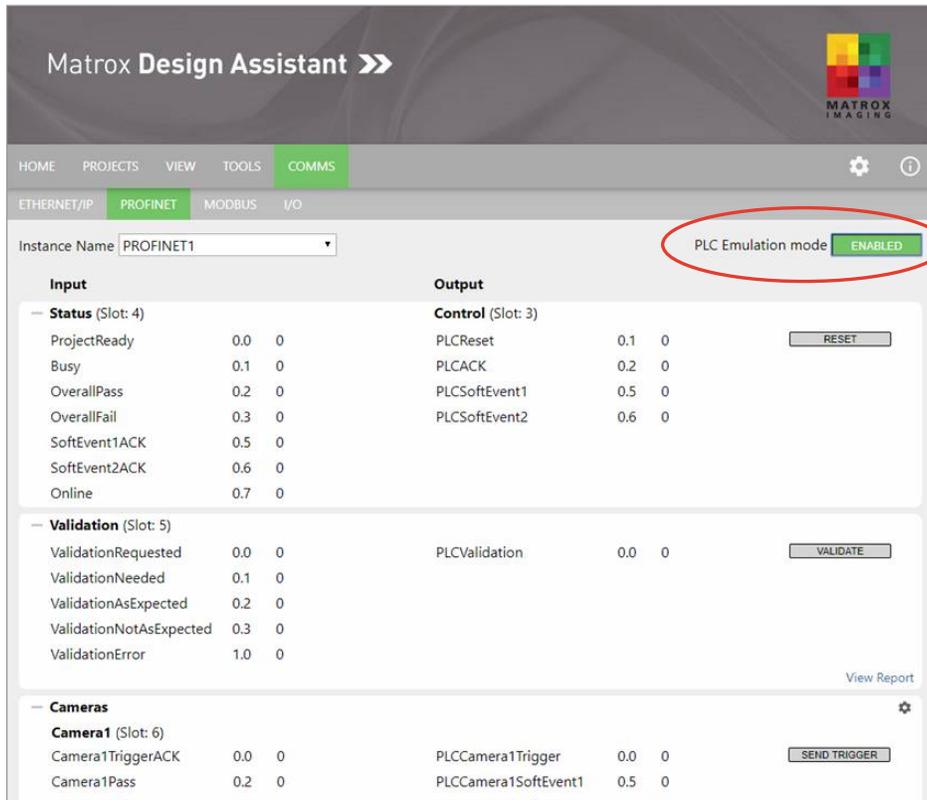


- 1 Programmable logic controller (PLC)**  
sends project validation request to vision system(s)
- 2 Vision system(s)**  
send(s) current project to validation server
- 3 Validation server**  
verifies project and returns status to vision system(s)
- 4 Vision system(s)**  
forward(s) validation status to PLC

# Utilities (cont.)

## PLC interface emulation

While developing a project in Aurora Design Assistant, the PLC interface emulator is used to test communication in instances when a physical one is not connected. Values can be changed and viewed dynamically to test the communication between the project and the PLC. The PLC interface emulator supports CC-Link IE Field Basic, EtherNet/IP<sup>2</sup>, MODBUS over TCP/IP, and PROFINET protocols for communication; these can be activated and controlled from the management portal.



# Connectivity

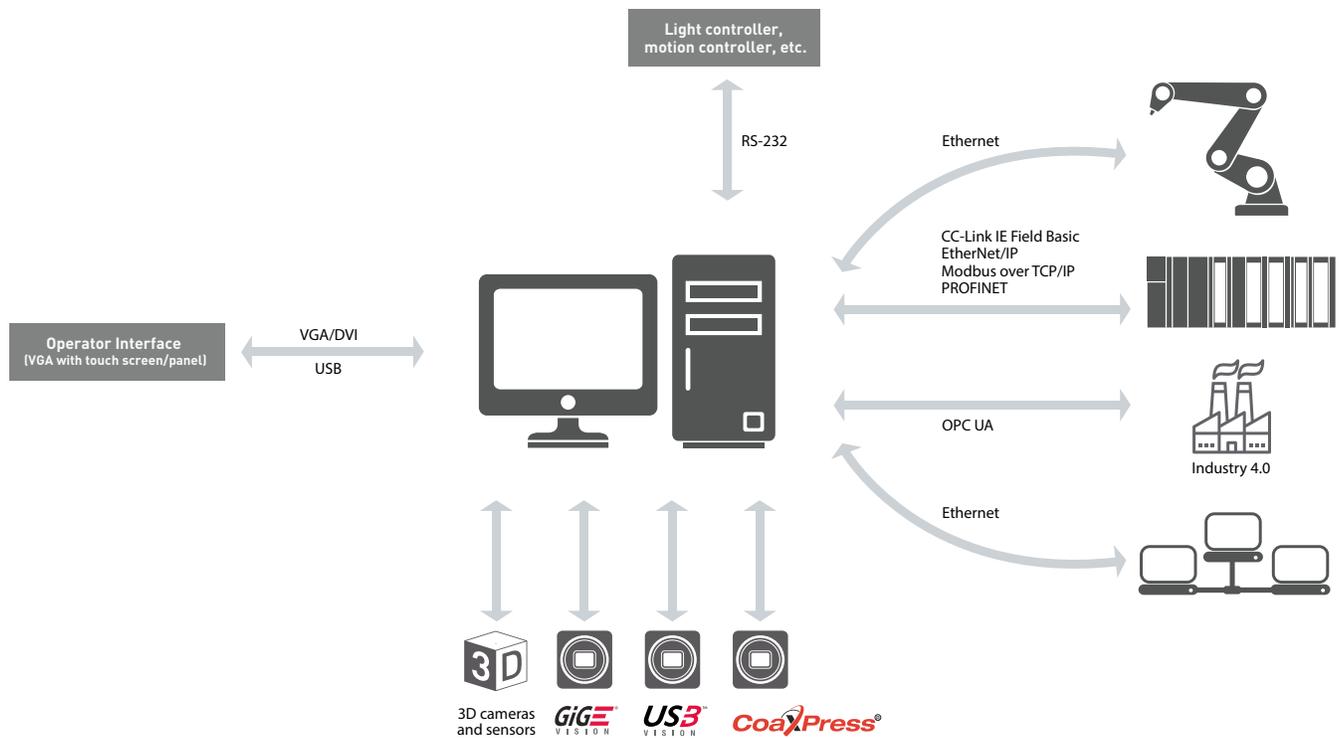
## Connect to devices and networks

Aurora Design Assistant can capture images from any CoaXPress, GigE Vision, or USB3 Vision compliant camera. Image capture from CoaXPress cameras happens with the use of a [Zebra Rapixo CXP](#) frame grabber. For GigE Vision cameras, the exact capture time can be obtained from IEEE 1588 timestamps.

The software can communicate over Ethernet networks using the TCP/IP as well as the CC-Link IE Field Basic, EtherNet/IP2, Modbus over TCP/IP, and PROFINET protocols, enabling interaction with programmable logic/automation controllers. Its QuickComm facility provides ready-to-go communication with these controllers. Aurora Design Assistant supports OPC UA communication for interaction with manufacturing systems and

direct communication with select robot controllers for 2D vision-guided robotic applications. Supported robot-controller makes and models currently include the ABB IRC5; DENSO RC8; Epson RC420+ and RC520+; Fanuc LRMate200iC and LRMate200iD; KUKA KR C2; and Stäubli CS8, CS8C HP, and CS9 controllers.

Aurora Design Assistant can be configured to interact with automation devices through a computer's COM ports. Aurora Design Assistant can also directly interact with the I/Os built into a Zebra [vision controller](#), [smart camera](#), and [I/O card](#) as well as the I/O available on a GigE Vision or USB3 Vision camera.



# Supported Environments

## For Windows

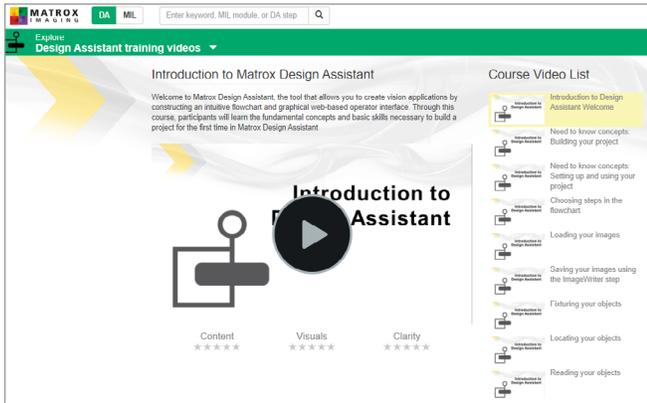
- 64-bit Windows 10 (versions 1809 to 21H2) and Windows 11 (version 21H2)

# Training and Support

## Vision Academy

Vision Academy provides all the expertise of live classroom training, with the convenience of on-demand instructional videos outlining how to get the most out of Aurora Design Assistant vision software. Available to customers with valid Aurora Design Assistant maintenance subscriptions, as well as those evaluating the software, users can seek out training on specific topics of interest, where and when needed.

Vision Academy aims to help users increase productivity, reduce development costs, and bring applications to market sooner. For more information, see Vision Academy.



## Professional Services

Professional Services delivers deep technical assistance and customized trainings to help customers develop their particular applications. These professional services comprise personalized training; assessing application or project feasibility (e.g., illumination, image acquisition, and vision algorithms); demo and prototype applications and/or projects; troubleshooting, including remote debugging; and video and camera interfacing.

Backed by the Vision Squad—a team of high-level vision professionals—Professional Services offer more in-depth support, recommending best methods with the aim of helping customers save valuable development time and deploy solutions more quickly. For more information on pricing and scheduling, contact Zebra sales.

## Aurora Design Assistant maintenance program

Aurora Design Assistant users have access to a Maintenance Program, renewable on a yearly basis. This maintenance program entitles registered users to free software updates and entry-level technical support from Zebra, as well as access to Vision Academy.

For more information, please refer to the Software Maintenance Programs.

# Ordering Information

Part number	Description
Aurora Design Assistant Development Package	
DAXWINPU	<p>Aurora Design Assistant development package for Windows. Includes installation media with IDE and online documentation. Also includes a single-user USB hardware license key for design-time environment and all run-time packages as well as a Aurora Design Assistant Maintenance registration number. Not required if ordering Zebra 4Sight EV6/XV6 or Zebra Iris GTX with Aurora Design Assistant.</p> <p>Note: 75% discount for DAXWINPU if purchased with a Aurora Imaging Library development package (i.e., MILXWINPU) for the same user with name as proof or for a Aurora Imaging Library user with valid maintenance with the registration number as proof. 50% educational discount for DAXWINPU with proof of affiliation with an academic institution. Discounts cannot be combined.</p>
Aurora Design Assistant Maintenance Program	
DAMAINTENANCE	<p>One-year extension to the Aurora Design Assistant maintenance program per developer.</p> <p>Note: Included in the original purchase price of the Aurora Design Assistant development package, Zebra 4Sight EV6/XV6, and Zebra Iris GTX with Aurora Design Assistant, it entitles registered users to one year of technical support and access to updates. 75% discount for DAMAINTENANCE if purchased with MILMAINTENANCE for the same user with name as proof. 50% educational discount for DAMAINTENANCE with proof of affiliation with an academic institution. Discounts cannot be combined.</p>
Vision Academy Online Training	
<i>Included with Aurora Design Assistant Maintenance Program</i>	<p>Vision Academy Online provides a range of categorized instructional videos on how to use the software to create applications. Vision Academy Online is available to customers with current Aurora Design Assistant maintenance subscriptions, as well as those evaluating the software.</p> <p>Visit <a href="http://www.matrox.com/imaging/en/vision_academy/">www.matrox.com/imaging/en/vision_academy/</a> for more information.</p>
Vision Academy On-Premises Training	
DATRAIN <i>Ask for availability.</i>	<p><i>Introduction to Aurora Design Assistant:</i> Three-day instructor-led training on developing machine vision applications using the Aurora Design Assistant IDE.</p> <p><i>Key topics:</i> Developing a vision or inspection system using flowcharts instead of coding; setting up an operator view; choosing analysis and processing tools; interfacing to automation controllers.</p> <p>Visit <a href="https://imaging.matrox.com/en/products/services/vision-academy/training/on-premises">https://imaging.matrox.com/en/products/services/vision-academy/training/on-premises</a> for more information.</p>
Aurora Design Assistant Run-time License Software Keys	
DXRTxxxxxxx00	<p>Aurora Design Assistant runtime software license key. The user must supply a lock code generated using the license management utility (MILConfig's Licensing tab) or the smart camera portal (Licensing tab on the Settings page). This unique lock code identifies the target system and package(s) to license. The correspondence between the below packages and flowchart steps comes from the Project License Information dialog accessible from the Platform menu in Aurora Design Assistant.</p> <p>Note: Combine packages by substituting 0 at the appropriate position x with the appropriate letter or other digit.</p>
DXRTA000000000	Aurora Design Assistant Image Analysis package. Enables BeadInspection, BlobAnalysis, Camera (with calibration), CameraFocus, EdgeLocator, Measurement, ImageCorrection, IntensityChecker, ImageProcessing (also see DAXRT...Q...), LoadImage (with calibration), Remap, and Mask steps. Also required for AlignPlane, Crop3D, ExtractProfile, and Volume3D steps.
DXRTM000000000	Aurora Design Assistant Machine Vision package. Includes ImageAnalysis package and enables PatternMatching step.
DXRT0I00000000	Aurora Design Assistant Identification package. Enables CodeReader and CodeGrade steps.
DXRT0C00000000	Aurora Design Assistant StringReader and SureDotOCR package (steps).
DXRT0200000000	Both DXRT0I00000000 and DXRT0C00000000.
DXRT00J0000000	Aurora Design Assistant Image Compression package. Required for Camera, ImageWriter, and LoadImage steps with image (de) compression.

# Ordering Information (cont.)

Part number	Description
DXRT000G000000	Aurora Design Assistant Geometric Model Finder package. Enables ModelFinder, RectangleFinder, CircleFinder, EllipseFinder, and SegmentFinder steps.
DXRT0000S00000	Aurora Design Assistant Interface package. Enables image acquisition using GenTL, GigE Vision, and USB3 Vision. Note: Not required for Zebra 4Sight GPm/EV6/XV6 and Zebra Iris GTR/GTX bundled with Aurora Design Assistant. Also see DXRT0000030000.
DXRT00000R0000	Aurora Design Assistant Registration package. Enables PhotometricStereo and HDR steps.
DXRT0000030000	Aurora Design Assistant 3D Calibration and Supplemental package. Enables 3D camera and sensor acquisition as well as AlignPlane (also see DXRTA.../DXRTM... and DAXRT...Y...), Crop3D (also see DXRTA.../DXRTM...), ExtractProfile (also see DXRTA.../DXRTM...), FillGaps, Project3D, and Volume3D (also see DXRTA.../DXRTM... and DAXRT...Y...) steps.
DXRT0000020000	Both DXRT00000R0000 and DXRT0000030000.
DXRT000000Y000	Aurora Design Assistant Metrology package (step). Also required for AlignPlane and Volume3D steps.
DXRT000000Q000	Aurora Design Assistant Color Analysis package. Enables ColorMatcher step. Also required for the ImageProcessing step's color distance/projection operation.
DXRT000000B000	Both DXRT000000Y000 and DXRT000000Q000.
DXRT0000000Z00	Aurora Design Assistant Industrial and Robot Communications functionality. Note: Not required for Zebra 4Sight GPm/EV6/XV6 and Zebra Iris GTR/GTX bundled with Aurora Design Assistant X.
DXRT0000000N00	Aurora Design Assistant Classification package. Enables CNNClassIndex and CNNClassMap steps.
DXRT0000000200	Both DXRT0000000Z00 and DXRT0000000N00.
DXRTM2B2B2B200	All Aurora Design Assistant packages. Enables all Aurora Design Assistant steps.
<b>Software License Hardware ID Key</b>	
MILRTIDCMC	Software license USB hardware fingerprint and storage. Note: Replaces Zebra hardware as the fingerprint used to generate the unique system code. DXRT..00 is still required.
<b>Aurora Design Assistant Runtime License Hardware Key</b>	
DXRTxxxxxxx00U	Pre-programmed Aurora Design Assistant runtime USB hardware license key that enables appropriate package(s). Note: See Aurora Design Assistant Runtime License Software Keys section for available selections. Alternative to DXRT..00.

Endnotes:

1. The software may be protected by one or more patents; see [www.matrox.com/patents](http://www.matrox.com/patents) for more information.
2. Certification pending.
3. Through an update.
4. With compute capability 3.5 to 8.6. Refer to <https://developer.nvidia.com/cuda-gpus> for correspondence to GPU models.



**NA and Corporate Headquarters**  
+1 800 423 0442  
inquiry4@zebra.com

**Asia-Pacific Headquarters**  
+65 6858 0722  
contact.apac@zebra.com

**EMEA Headquarters**  
zebra.com/locations  
contact.emea@zebra.com

**Latin America Headquarters**  
zebra.com/locations  
la.contactme@zebra.com