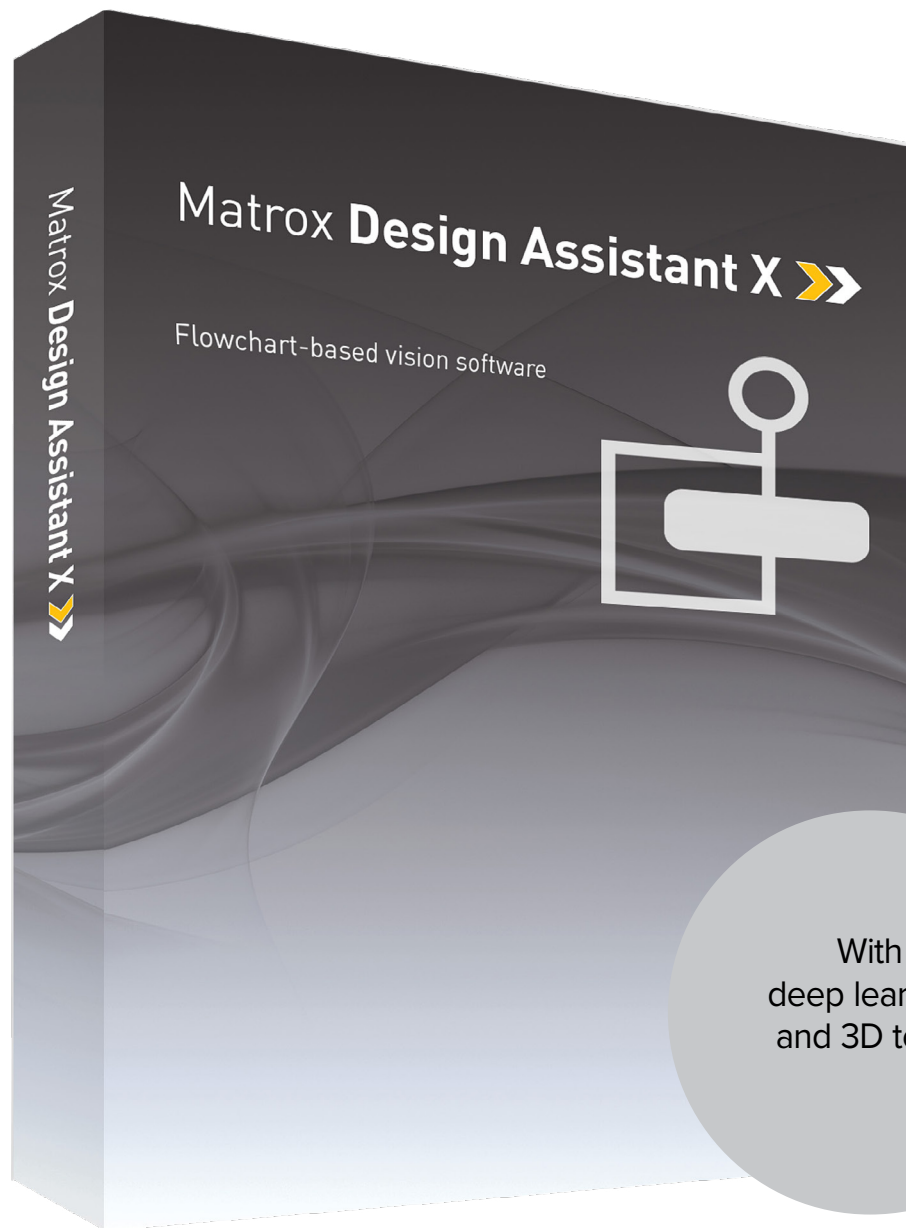


Zebra Aurora Design Assistant™

Intuitive, Versatile, and Extendable Integrated Development
Environment for Machine Vision Applications



Overview

Flowchart-based vision software

Aurora Design Assistant¹, 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 [Iris GTX 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 [Iris GTX 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, detection, 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 [Iris GTX 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™2, 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

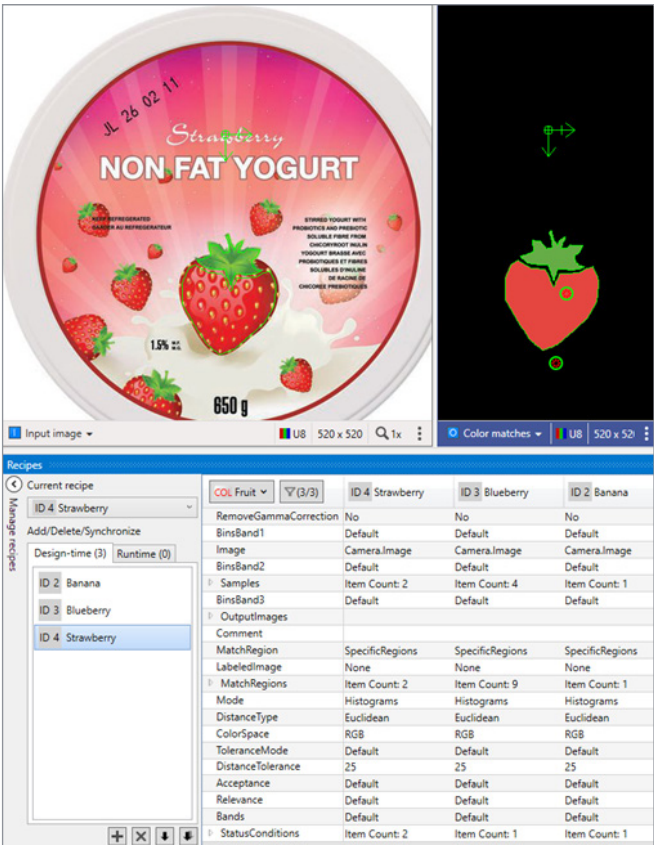
Increase productivity and reduce development costs with Vision Academy [video-based](#) and [live](#) training

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.



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, tables 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.

Latest key additions and enhancements

Object detection using deep learning

3D surface matcher

3D box finder

Alternate barcode and DataMatrix readers

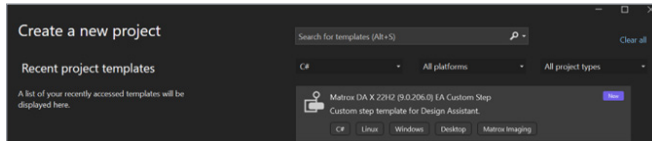
Project differences utility

User authentication

Overview (cont.)

Custom flowchart steps

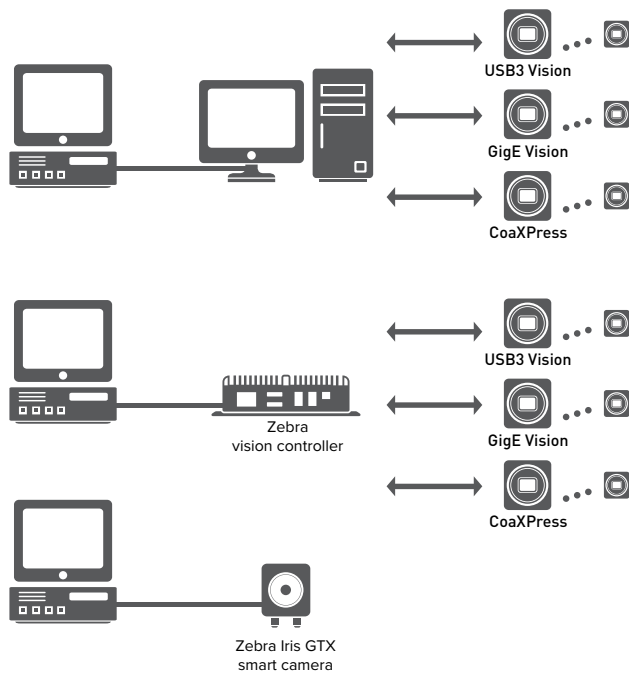
Users have the ability to extend the capabilities of Aurora Design Assistant by way of the included Custom Step software development kit (SDK). The SDK, in combination with Microsoft Visual Studio® 2022, enables the creation of custom flowchart steps using the C# programming language. These steps can 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 Iris GTX smart camera.



Deployment options

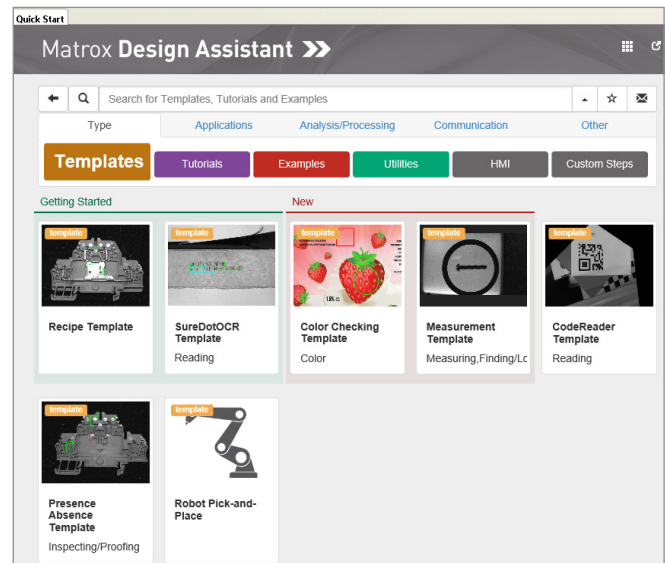
Project templates for quicker start-up

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.

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

Get quick access to contextual documentation

The screenshot displays the Aurora Design Assistant IDE interface. On the left, a flowchart titled 'MainFlowchart' shows a sequence of steps: 'Camera', 'CircleFinder', 'Volume3D', 'CheckIfBroken', and 'Status'. The central workspace features two 3D visualizations: a red and yellow volume rendering on the left and a blue and green volume rendering on the right. Below these, a configuration panel for 'Volume3D' is visible, showing parameters like 'Region mode', 'Shape', and 'Radius'. On the right side, a 'Quick Access' panel provides contextual documentation for the 'Volume3D' step, including a description of its function and a 'Step Log' showing the execution flow.

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

Volume status	Volume	Execution Time
WarningGaps	26537	0.0583701 s
WarningGaps	27452	0.0576217 s
WarningGaps	18738.6	0.0473563 s
WarningGaps	11865.7	0.0399821 s
WarningGaps	26217.3	0.0576251 s
WarningGaps	26256.4	0.0570474 s
WarningGaps	18747.5	0.0448805 s
WarningGaps	25580.5	0.0550637 s
WarningGaps	26527	0.0580562 s
WarningGaps	27453.2	0.0588662 s
WarningGaps	18738.6	0.0464841 s

Operator View

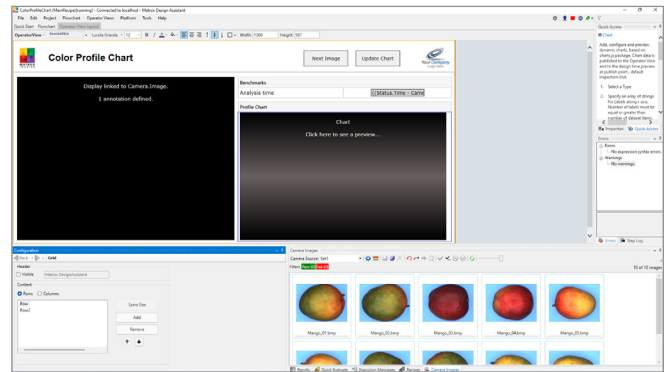
Operator interface viewable anywhere

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 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 [Iris GTX 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.

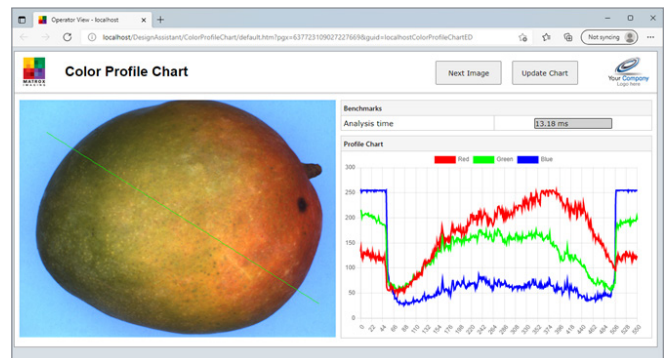
A stand-alone HMI application can be created using Microsoft Visual Studio to run on the local or a remote computer as an alternative to the web-based operator interface.

Security features

Access to specific Operator Views and the Aurora Design Assistant management portal can be made to require user authentication (i.e., username and password) so only authorized personnel can modify key settings in a deployed project. The access control is implemented using the [Keycloak](#) identity and access management service with support for integration with the Microsoft Active Directory service. A project can be locked to a specific Zebra [Iris GTX 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

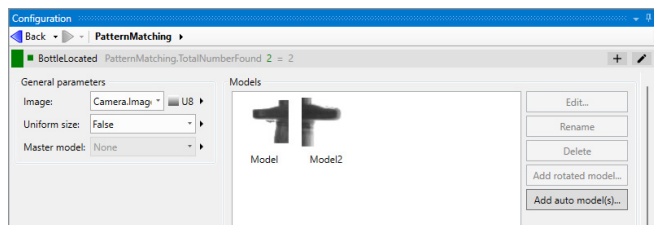
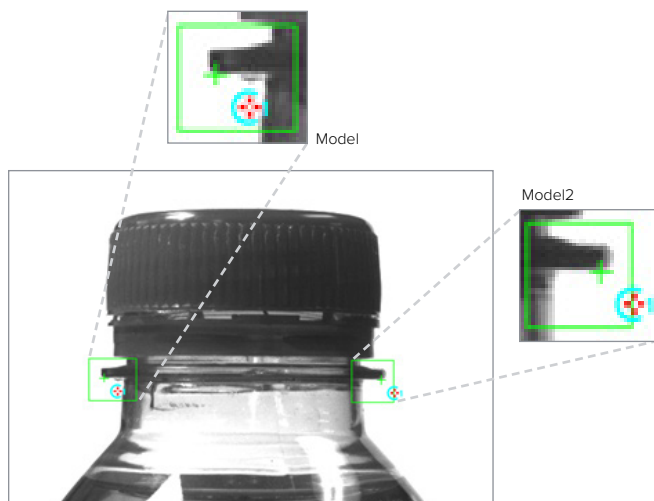
Image analysis and processing

Central to Aurora Design Assistant are flowchart steps for calibrating, enhancing, and transforming images; locating objects; extracting, analyzing, 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

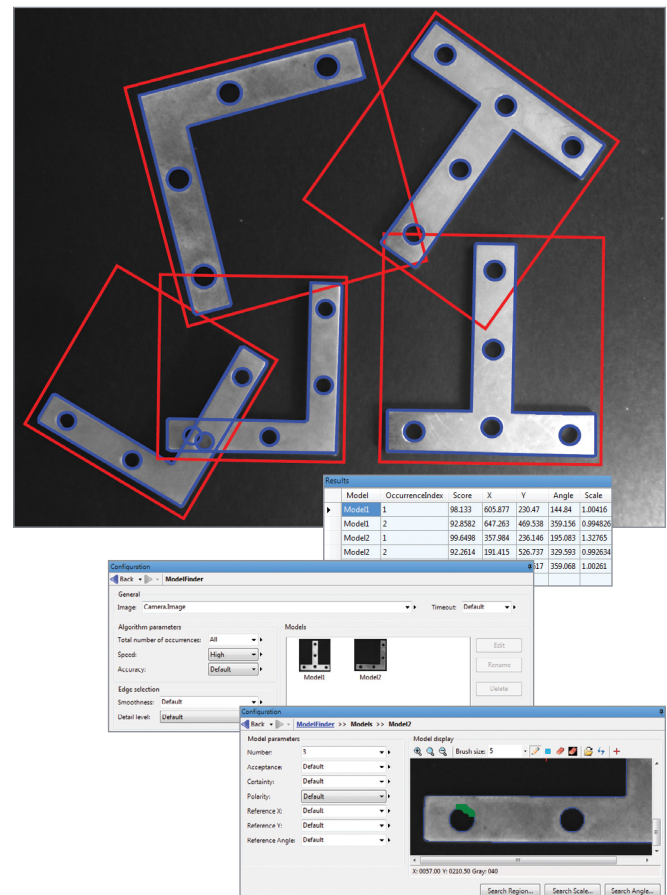
Aurora Design Assistant includes two steps for performing pattern recognition: PatternMatching and ModelFinder. These steps are primarily used to locate complex objects for guiding a gantry, stage, or robot, or for directing subsequent analysis and processing steps.

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.



PatternMatching step

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.



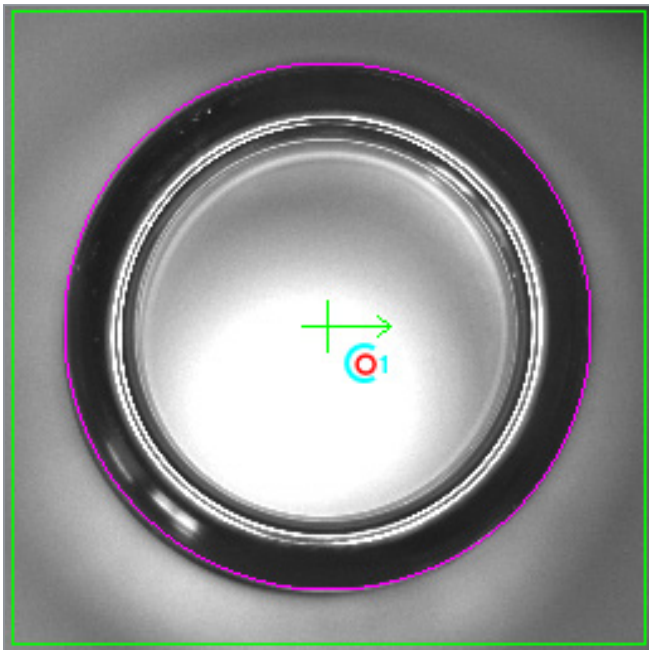
ModelFinder step

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.

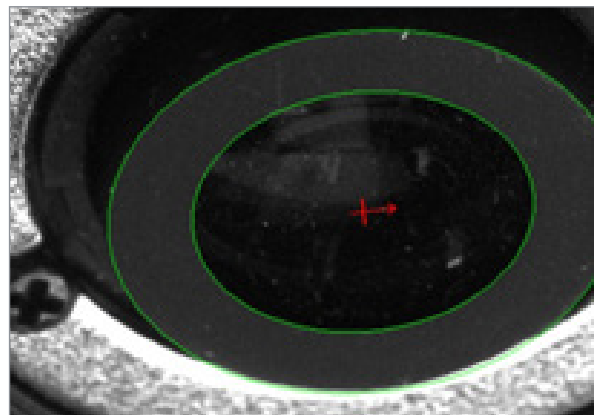
The shape-finding step computes the total number of found occurrences; for each occurrence, the tool can provide the center position and score relative to the reference. It can also give the radius and scale for circles; the angle, aspect ratio, width, and scale for ellipses and rectangles; and the start 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

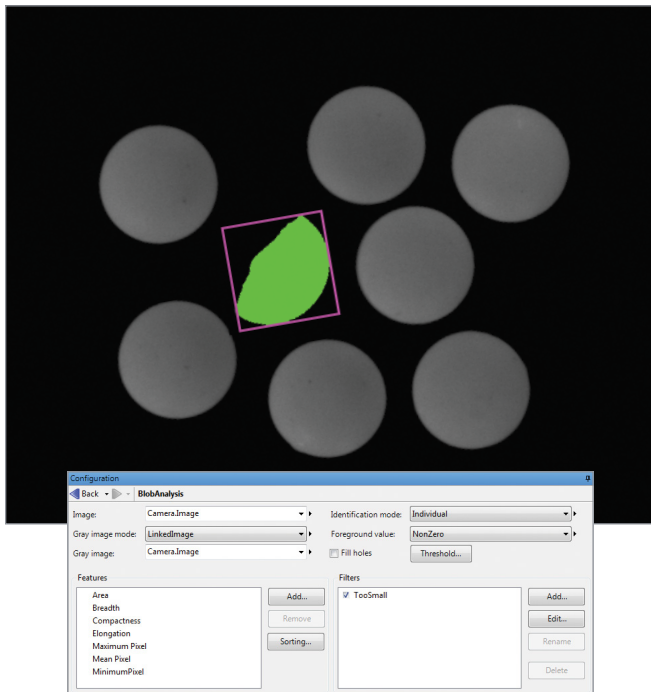


EllipseFinder step

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

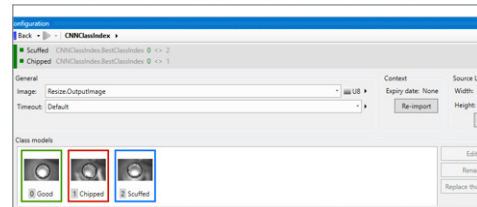
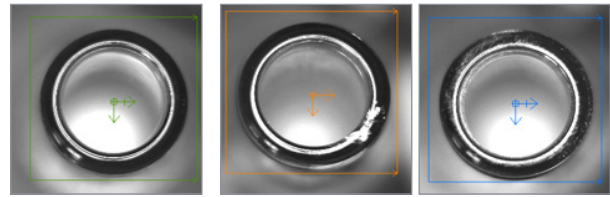
Aurora Design Assistant includes classification steps for automatically categorizing image content using machine learning. These steps make use of deep learning technology—specifically the convolutional neural network (CNN) and variants—in three distinct approaches.

The 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.

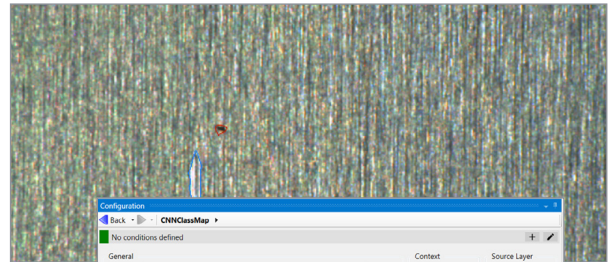
The segmentation approach—implemented by the CNNClassMap step—generates maps indicating the pre-established class and score for all image pixels. It is appropriate for detection tasks where the objective is to determine the incidence, location, and extent of flaws or features. These features can then be further analyzed and measured using traditional tools like the BlobAnalysis step.

The object detection approach—implemented by the CNNObjectDetect step—locates instances of pre-established classes. It is suited for inspection tasks whose goal is to find, size and count objects or features. The result for each located instance is the most likely class, the score and a bounding box including the corner coordinates, center, height and width. These classification steps are particularly well suited for analyzing images of highly textured, naturally varying, and acceptably deformed goods in complex and varying scenes.

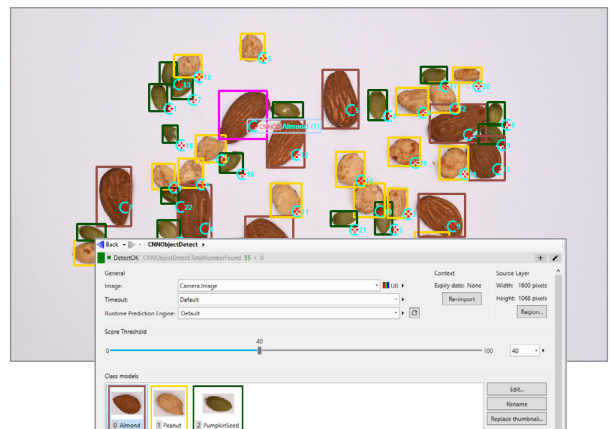
Users can opt to train a deep neural network on their own—using the included [Aurora Imaging CoPilot](#) application—or commission Zebra to do so using previously collected images that are both adequate in number and representative of the expected application conditions. Different types of training are supported, such as transfer learning and fine-tuning, all starting from one of the supplied pre-defined deep neural network architectures. Aurora Imaging 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 GPU³ or x64-based CPU. Inference is performed on a CPU, Intel integrated GPU³, and NVIDIA GPU³, supporting different speed and budget requirements.



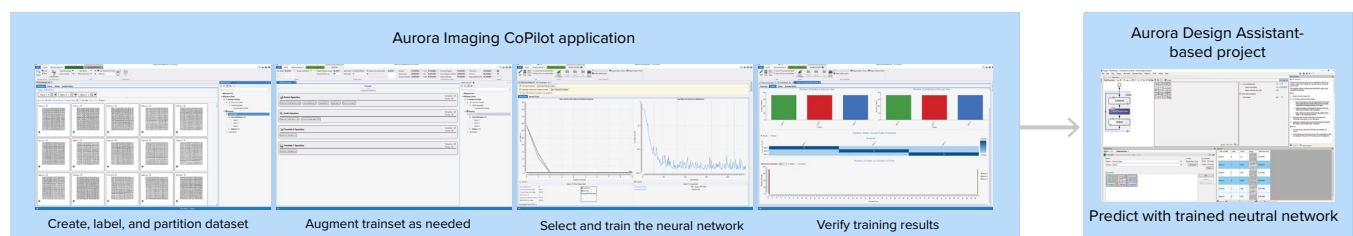
CNNClassIndex step



CNNClassMap Step



CNNObjectDetection Step



Aurora Imaging CoPilot workflow for Classification training

Field-Proven Vision Tools (cont.)

1D and 2D measurement steps

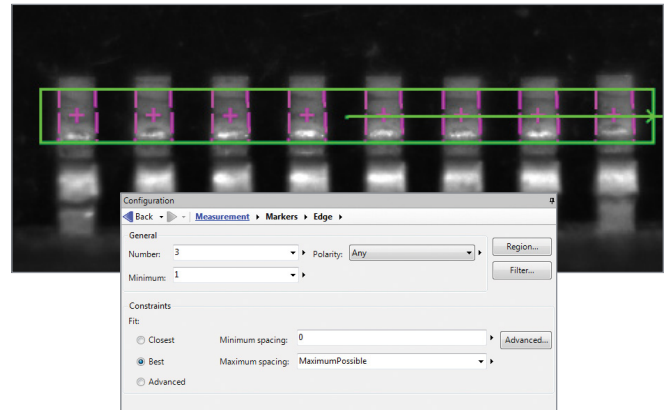
Aurora Design Assistant includes three steps for measuring: Measurement, BeadInspection, and Metrology. These tools are predominantly used to assess manufacturing quality.

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

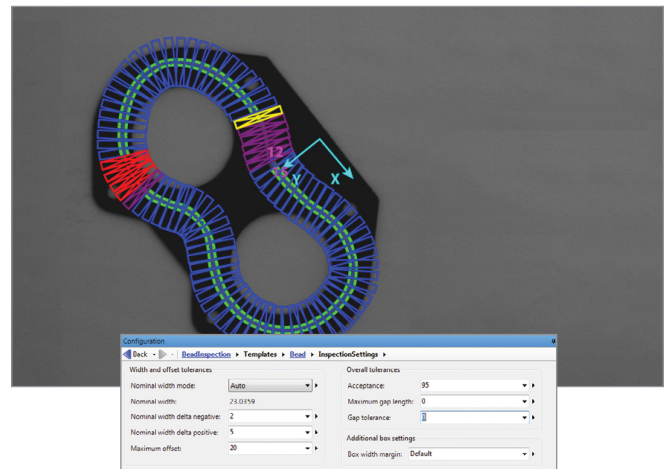
The BeadInspection step is for inspecting material that is applied as a continuous sinuous bead, such as adhesives and sealants, or its retaining channel. The step identifies discrepancies in length, placement, and width, as well as discontinuities. The BeadInspection step works by accepting a user-defined coarse 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.

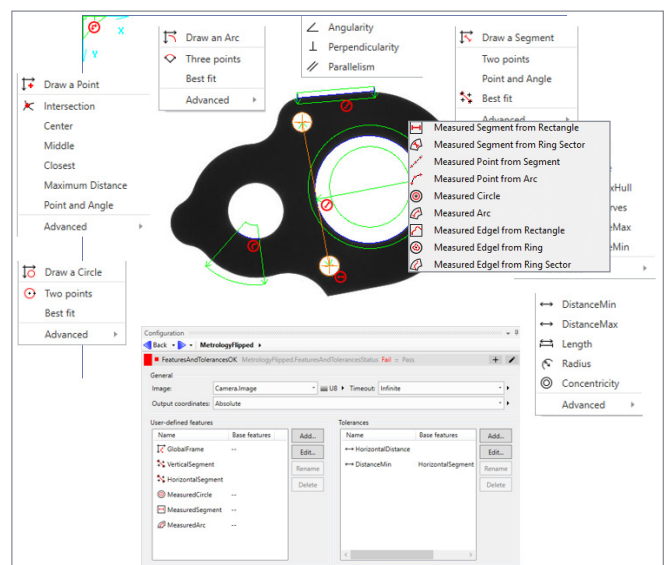
The Metrology step's effectiveness is maintained when subject to uneven changes in scene illumination, which relaxes lighting requirements. The expected measured and constructed geometric features, along with the tolerances, are kept together in a template which is easily repositioned using the results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



Measurement step



BeadInspection step



Metrology step

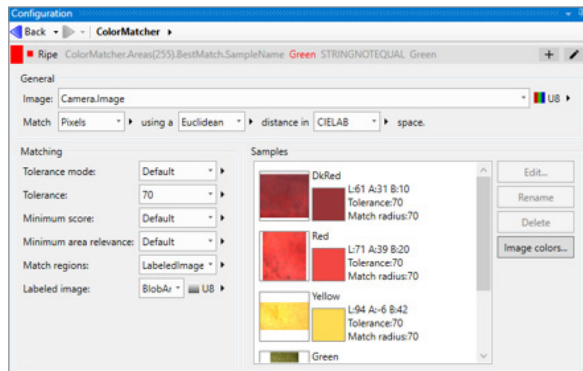
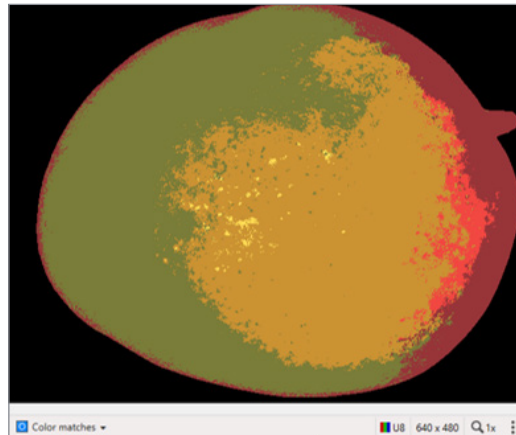
Field-Proven Vision Tools (cont.)

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.)

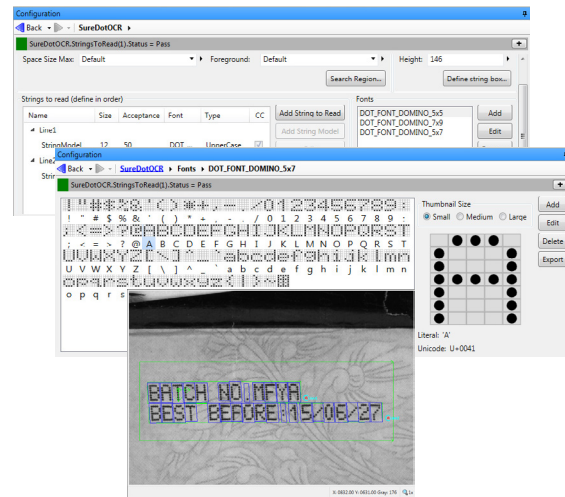
Character recognition steps

Aurora Design Assistant provides three steps for character recognition: SureDotOCR, StringReader, and Deep Learning OCR. 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.

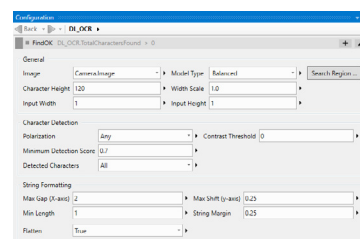
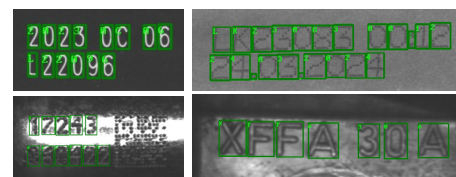
The Deep Learning OCR (DL_OCR) step uses a pre-trained deep neural network to read Latin-based characters, including punctuation marks and certain symbols, without the need to provide or teach it specific fonts. The tool automatically handles dot-formed and solid text that is printed, engraved, and embossed on a surface. Set up is straightforward, typically only needing to specify the average expected character height. It can read text at modest angles from the image horizontal and at any angle using a region established from a prior location operation.



SureDotOCR step



StringReader step



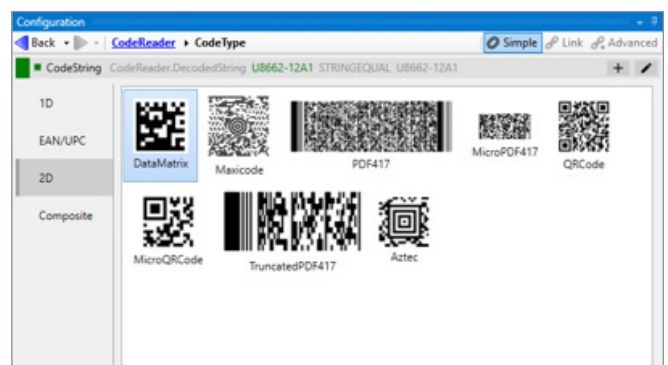
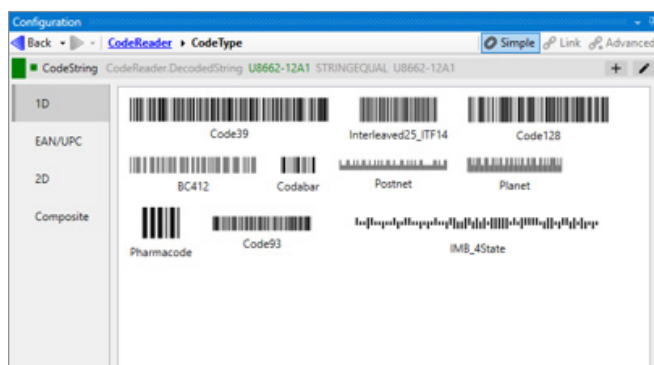
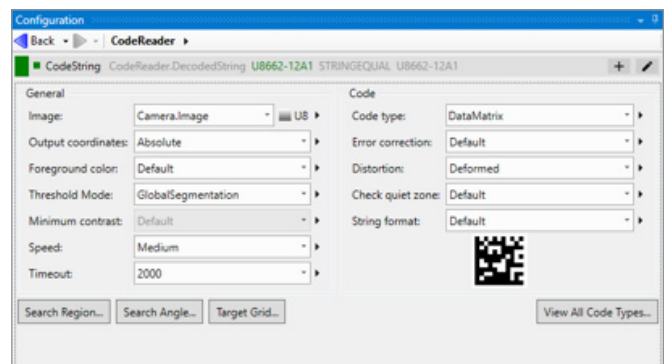
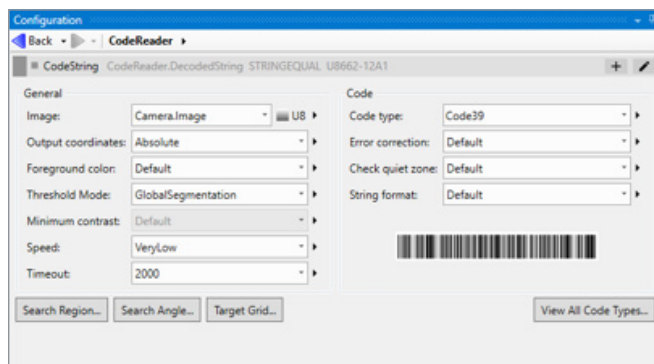
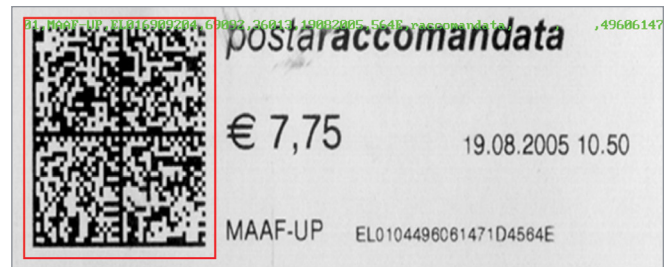
DL_OCR Step

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. The alternate CodeReaderIK and DataMatroxAVL steps provide the ability to read multiple codes at once among other traits.

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



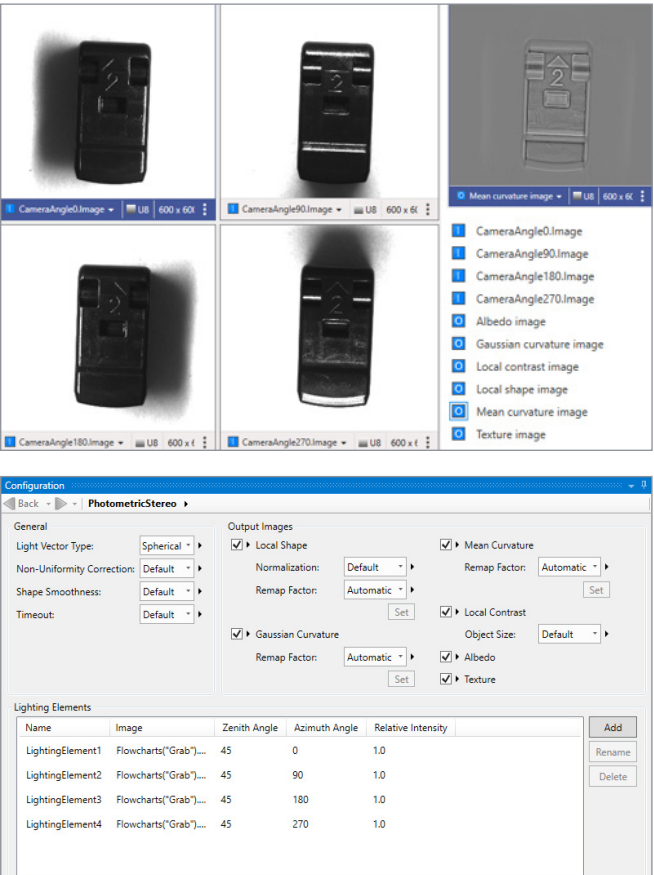
CodeReader step (1D)

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

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).



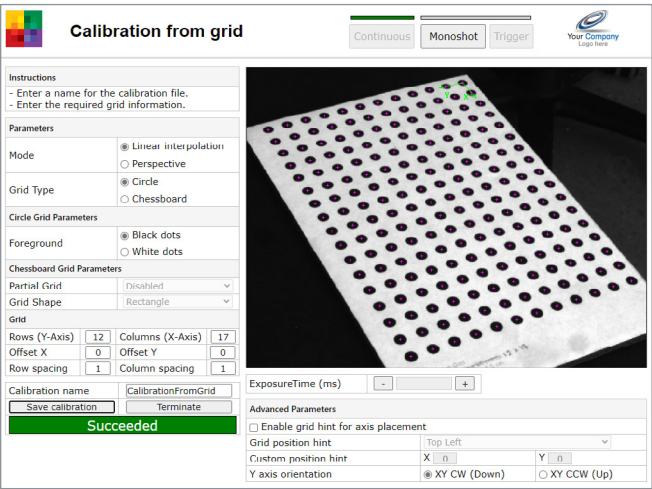
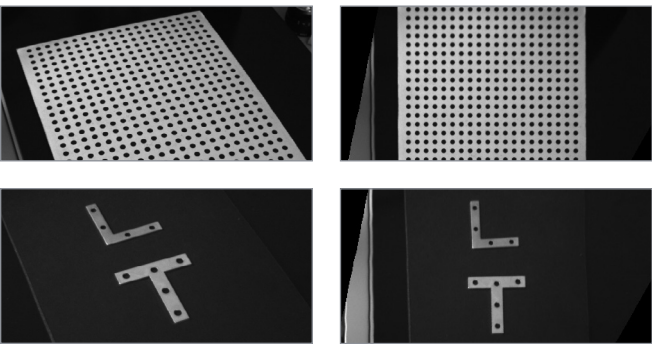
HDR step

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.

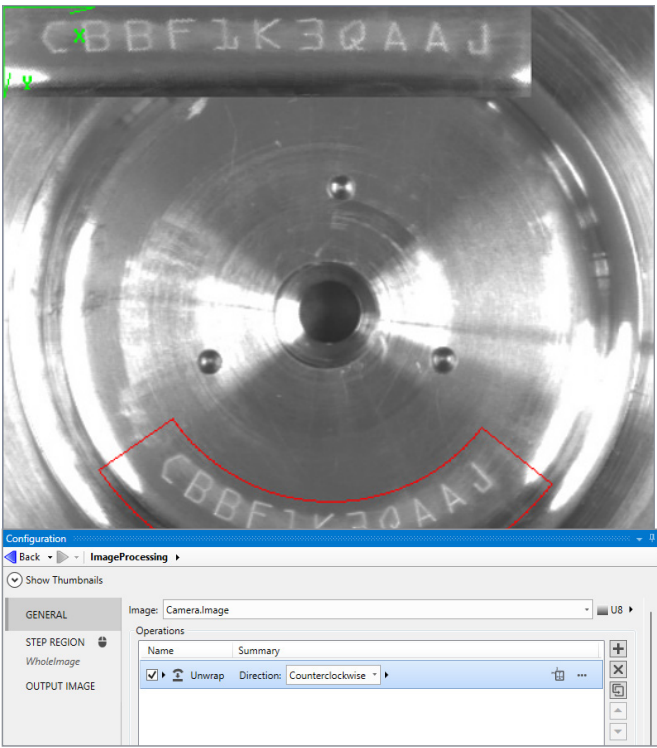


Calibration utility

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.)

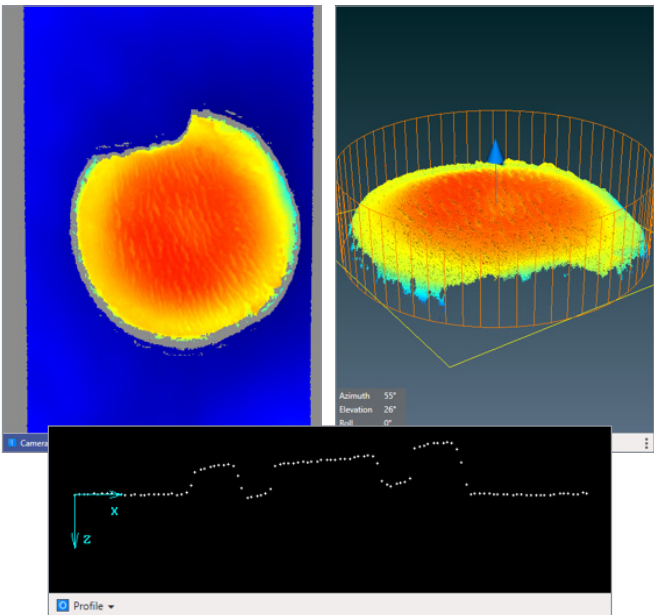
3D acquisition and display

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. Also available is 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

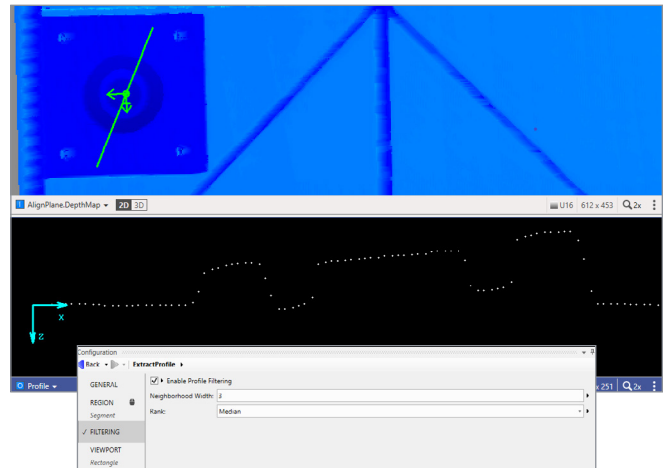
Manufacturer	Model(s)
Automation Technology	C5-CS series
Arcure (Blaxtair)	Omega
Basler	blaze
FRAMOS	D400e series
Intel	D400 series
LMI	Gocator 2000 and 3000 series
Lucid	Helios(2/2+)
Micro Epsilon	scanCONTROL and surfaceCONTROL
Photoneo	PhoXi and MotionCam-3D series
SmartRay	ECCO series
Wenglor	weCat3D 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.

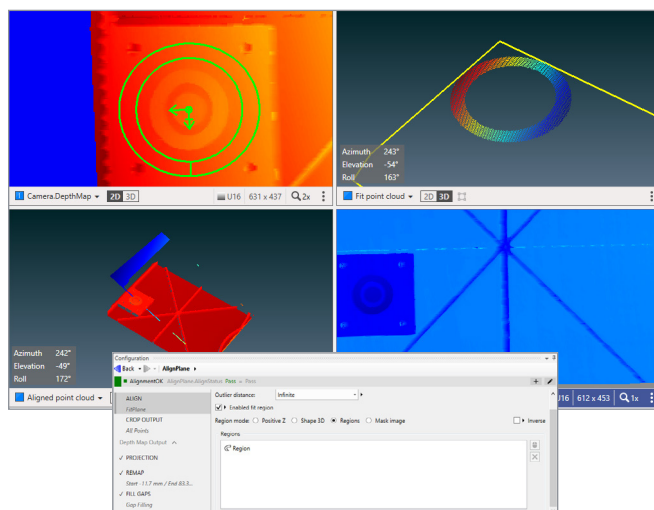
Field-Proven Vision Tools (cont.)

3D analysis and processing

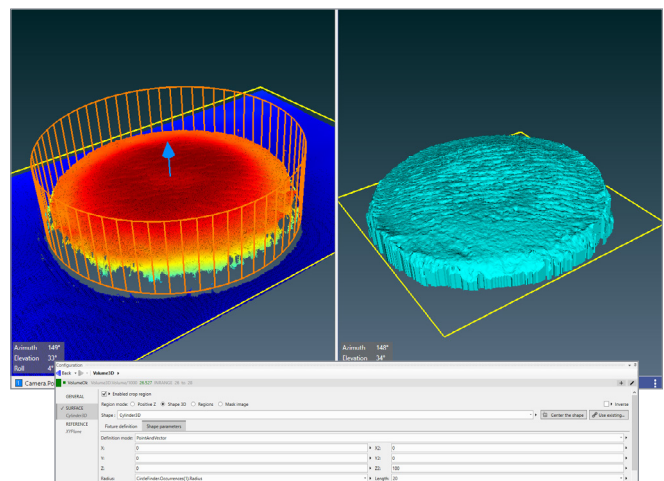
Aurora Design Assistant is able to inspect the data produced by 3D cameras and sensors using the previously mentioned 2D or additional 3D-specific analysis and processing steps. The ExtractProfile step extracts a profile from a depth map or point cloud for analysis using the Metrology step. The Project3D 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. The 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. The PlaneFit step creates a reference plane in a depth map or point cloud. The AlignPlane step rotates a scene to align a reference plane with the Z axis origin. The 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. The Subsample3D, Mesh3D, and Resample3D steps subsample a point cloud, mesh a point cloud, and resample a meshed point cloud respectively, typically in advance of a volume calculation. The Volume3D step is available to compute the volume of a depth map as well as a point cloud whose surface is meshed and filled.



ExtractProfile step



AlignPlane step

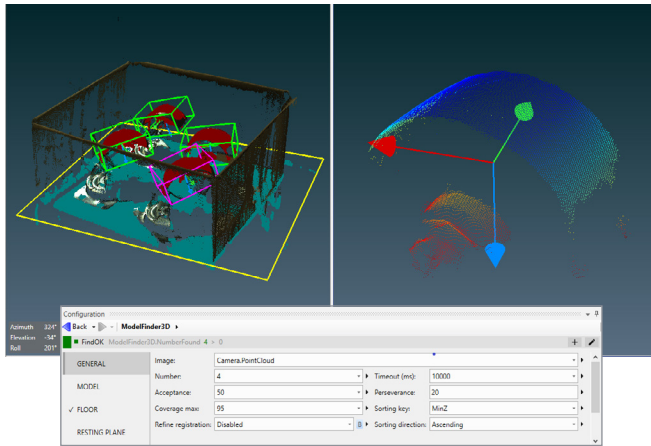


Volume3D step

Field-Proven Vision Tools (cont.)

3D surface matcher

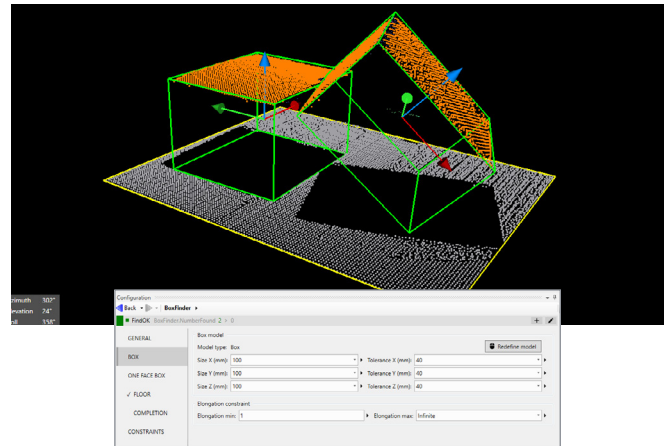
Aurora Design Assistant includes the ModelFinder3D step for finding a surface model— including multiple occurrences—at wide-ranging orientations in a point cloud. A surface model is defined from a point cloud obtained from a 3D camera or sensor, or from a CAD (PLY or STL) file. Various controls are provided to influence the search accuracy, robustness, and speed. Search results include the number of occurrences found and for each occurrence, the scores, center coordinates, estimated pose, and number of points.



ModelFinder3D step

3D box finder

Aurora Design Assistant features the BoxFinder step for locating one or more box-like objects in a point cloud. The boxes to find are specified graphically or numerically. The step returns the position and orientation information of a whole box and its faces while also providing dimensions. BoxFinder handles situations where a restricted number of faces are visible.

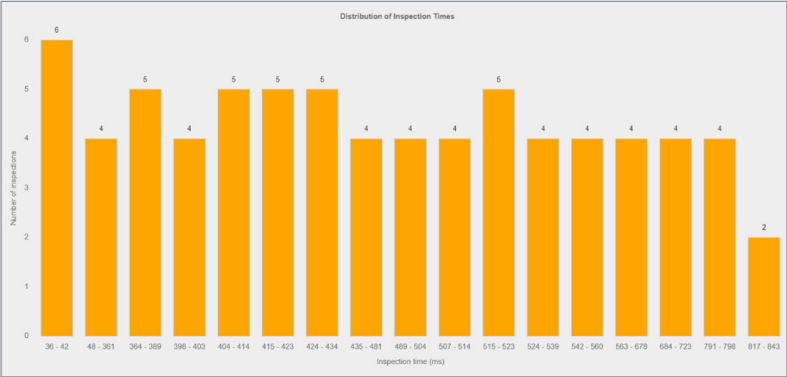
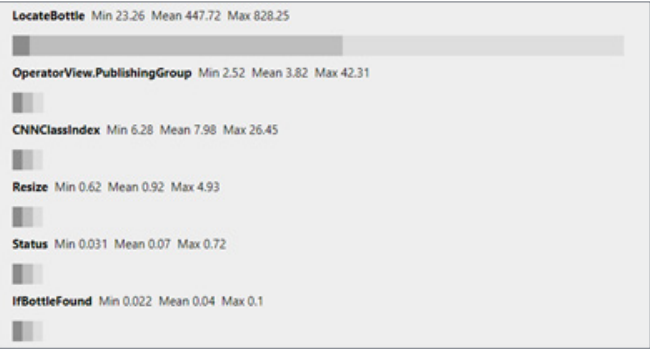
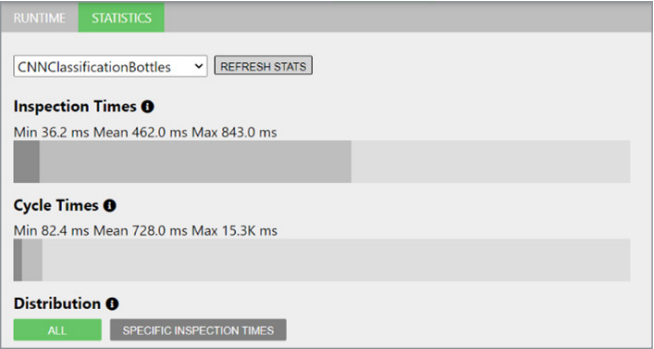
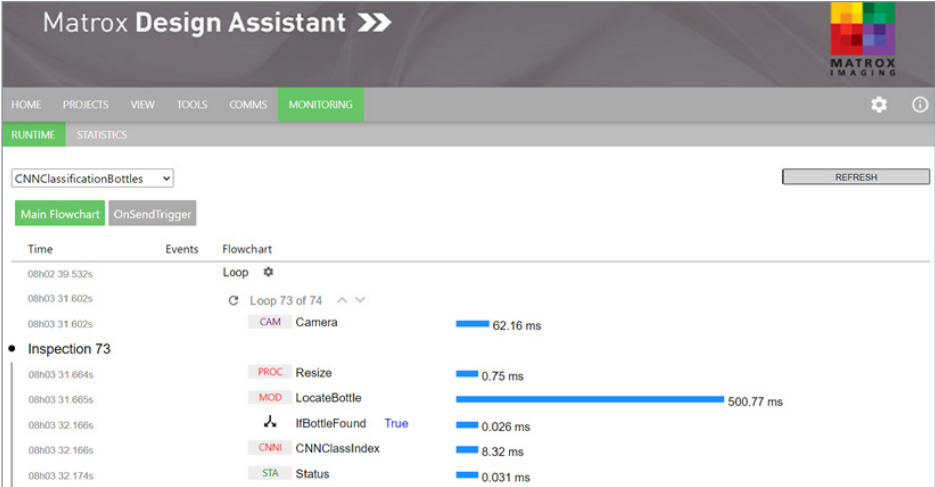


BoxFinder step

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 as 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.



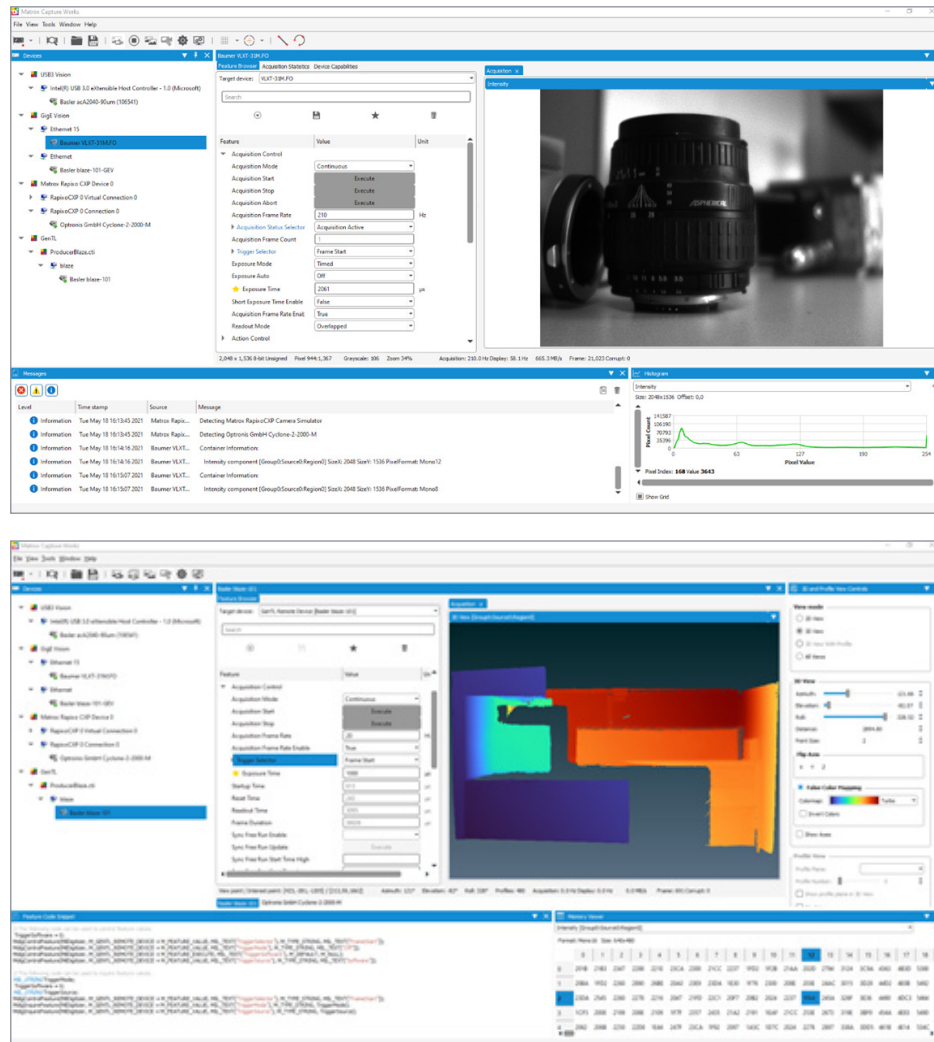
Runtime monitor

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 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.



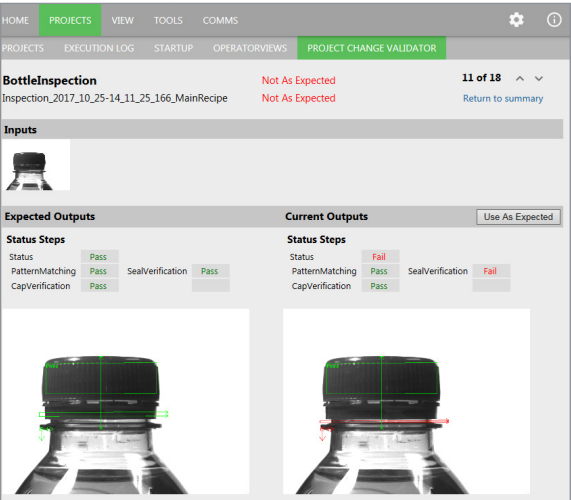
Aurora Capture Works configuration and test tool for GenICam-based interface standards

Utilities (cont.)

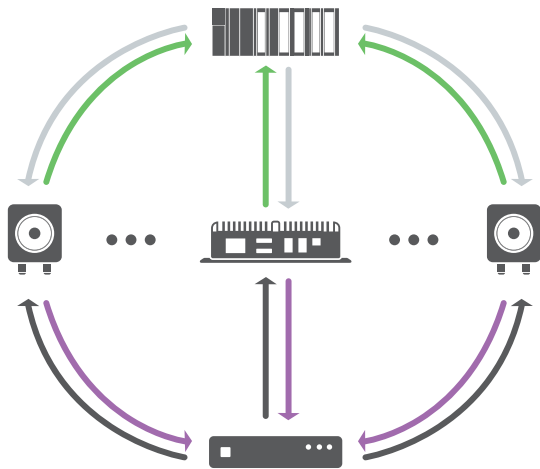
Project Change Validator

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.



Project Change Validator (view from management portal)

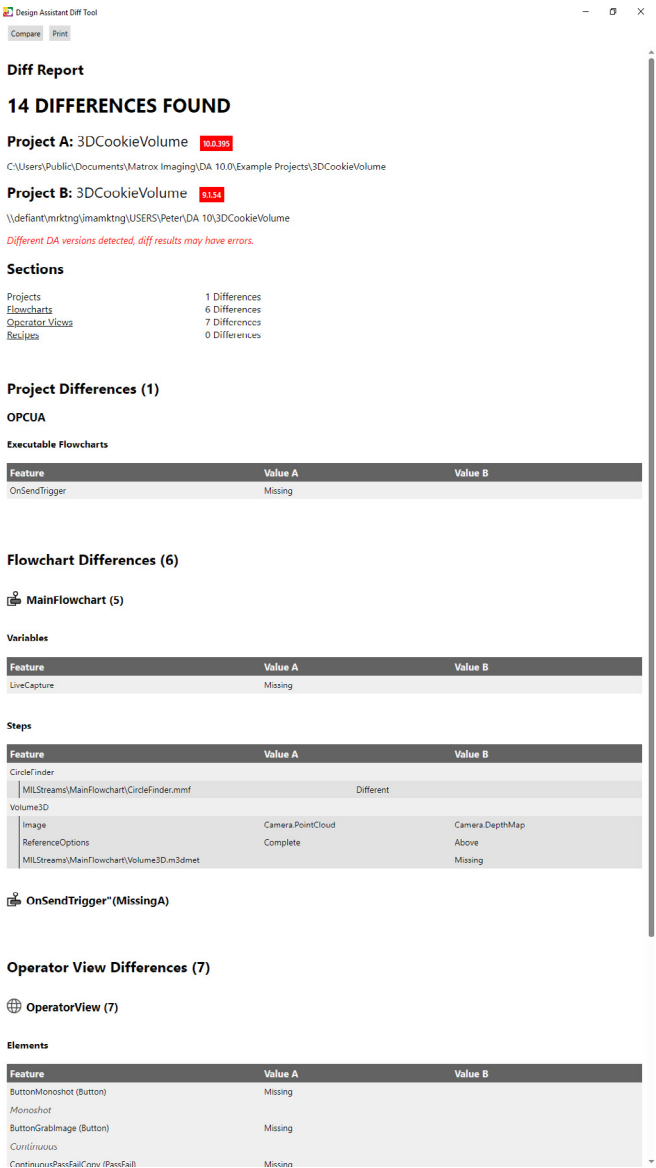


- 1 **Programmable logic controller (PLC)**
sends project validation request to vision system(s)
- 2 **Vision system(s)**
send(s) current project to validation server
- 3 **Validation server**
verifies project and returns status to vision system(s)
- 4 **Vision system(s)**
forward(s) validation status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

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 differences

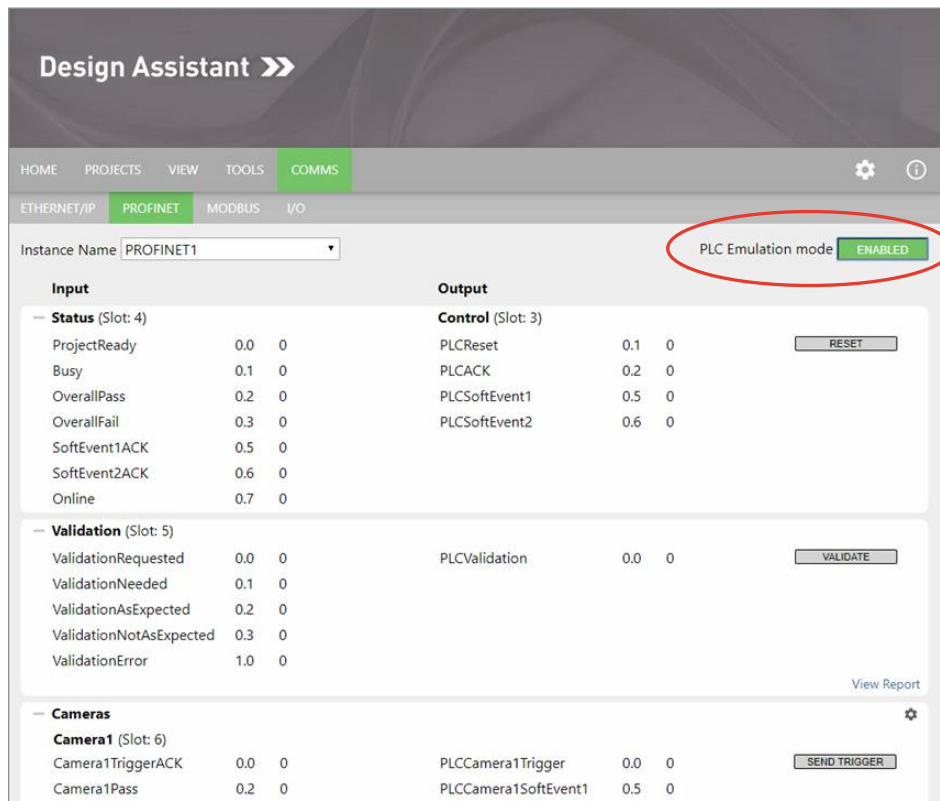
The project differences utility compares two projects: A deployed one against a main reference, two from different deployments, or two created using different versions of Aurora Design Assistant. The utility reports differences to various project elements including flowcharts, steps, recipes, and Operator Views.



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/IP², MODBUS over TCP/IP, and PROFINET protocols for communication; these can be activated and controlled from the management portal.



The screenshot shows the 'Design Assistant' interface with the 'COMMS' tab selected. The 'PROFINET' sub-tab is active. The 'Instance Name' is set to 'PROFINET1'. A red circle highlights the 'PLC Emulation mode' toggle, which is currently 'ENABLED'. The interface displays three sections: 'Status (Slot: 4)', 'Validation (Slot: 5)', and 'Cameras'. Each section contains a table of variables with their addresses and values, along with control buttons like 'RESET', 'VALIDATE', and 'SEND TRIGGER'.

Input			Output		
Status (Slot: 4)			Control (Slot: 3)		
ProjectReady	0.0	0	PLCReset	0.1	0
Busy	0.1	0	PLCACK	0.2	0
OverallPass	0.2	0	PLCSoftEvent1	0.5	0
OverallFail	0.3	0	PLCSoftEvent2	0.6	0
SoftEvent1ACK	0.5	0			
SoftEvent2ACK	0.6	0			
Online	0.7	0			
Validation (Slot: 5)					
ValidationRequested	0.0	0	PLCValidation	0.0	0
ValidationNeeded	0.1	0			
ValidationAsExpected	0.2	0			
ValidationNotAsExpected	0.3	0			
ValidationError	1.0	0			
Cameras					
Camera1 (Slot: 6)					
Camera1TriggerACK	0.0	0	PLCCamera1Trigger	0.0	0
Camera1Pass	0.2	0	PLCCamera1SoftEvent1	0.5	0

Connectivity

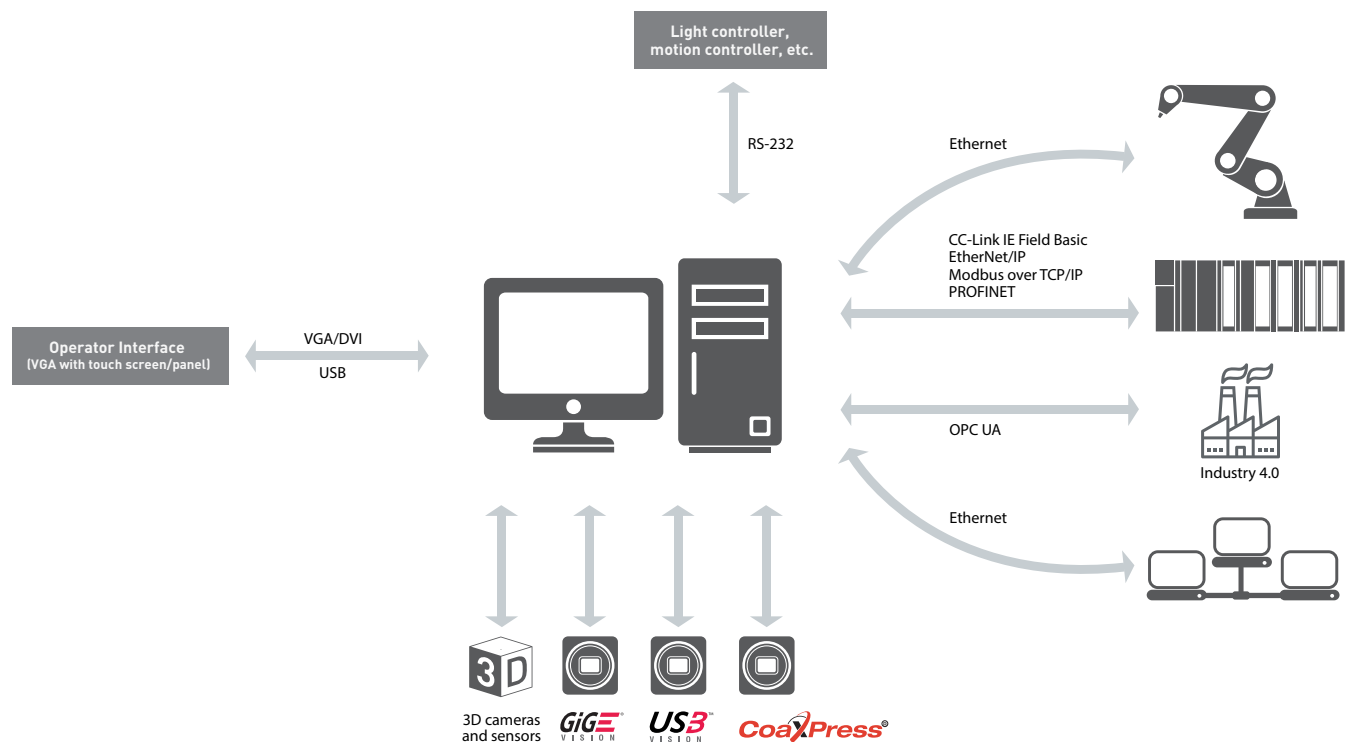
Connect to devices and networks

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](#), [Iris GTX 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

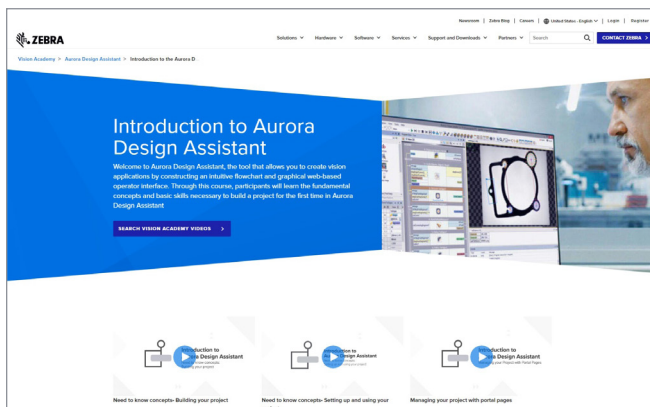
- 64-bit Windows 10 (versions 1809 to 22H2) and Windows 11 (version 21H2 and 22H2)

Training and Support

Vision Academy

[Vision Academy](#) provides all the expertise of live 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 plans, 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	Aurora Design Assistant 10 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.
Aurora Design Assistant Maintenance Program	
Z1R5-MVADAX-1000	One year extension to the Aurora Design Assistant maintenance program per developer.
Vision Academy Video-based Training	
<i>Included with Aurora Design Assistant Maintenance Program</i>	Vision Academy video-based training provides a range of categorized instructional videos on how to use the software to create applications. It is available to customers with current Aurora Design Assistant maintenance plans, as well as those evaluating the software. Visit https://www.zebra.com/us/en/vision-academy.html for more information.
Vision Academy Live Training	
<i>Ask for details including availability</i>	Vision Academy is a regularly scheduled series of live in-person and online intensive courses covering all aspects of using Aurora Design Assistant to develop vision applications. Each course, whether in-person or online, combines lectures delivered by support and/or development staff with hands-on labs to get attendees going with either software environment. Courses are available upon enrollment; tailored courses at Zebra headquarters or users' site can also be organized upon assessment. Visit https://www.zebra.com/us/en/vision-academy/live-online-and-classroom-based-training.html for more information.
Aurora Design Assistant Run-time License Software Keys	
DXRTxxxxxxxx00	Aurora Design Assistant 10 runtime software license key. The user must supply a lock code generated using the license management utility (MILConfig's Licensing tab) or the Zebra Iris GTX 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. Note: Combine packages by substituting 0 with the appropriate letter or other other digit at the appropriate position.
DXRTA000000000	Aurora Design Assistant 10 Image Analysis (IA) 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, BoxFinder, ExtractProfile, FillGaps, Mesh3D, ModelFinder3D, PlaneFit, Project3D, Resample3D, Subsample3D, and Volume3D steps.
DXRTM000000000	Aurora Design Assistant 10 Machine Vision (MV) package. Includes ImageAnalysis package and enables PatternMatching step.
DXRT0I00000000	Aurora Design Assistant 10 Identification package (ID). Enables CodeReader and CodeGrade steps.
DXRT0C00000000	Aurora Design Assistant 10 StringReader (SR) and SureDotOCR package (steps).
DXRT0200000000	Both DXRT0I00000000 and DXRT0C00000000.
DXRT00J0000000	Aurora Design Assistant 10 Image (De) Compression (Codec) package. Required for Camera, ImageWriter, and LoadImage steps with image (de)compression.

Ordering Information (cont.)

Part number	Description
DXRT000G000000	Aurora Design Assistant 10 Geometric Model Finder (GMF) package. Enables ModelFinder, RectangleFinder, CircleFinder, EllipseFinder, and SegmentFinder steps. Also required for BoxFinder and ModelFinder3D steps.
DXRT0000S00000	Aurora Design Assistant 10 Interface package. Enables image acquisition using GenTL, GigE Vision, and USB3 Vision. Note: Not required for Zebra 4Sight EV6/EV7/XV6 and Zebra Iris GTX bundled with Aurora Design Assistant. Also see DXRT0000030000.
DXRT00000R0000	Aurora Design Assistant 10 Registration package. Enables PhotometricStereo and HDR steps.
DXRT0000030000	Aurora Design Assistant 10 3D Calibration and Supplemental (3D) package. Enables 3D camera and sensor acquisition as well as AlignPlane (also see DXRTA.../DXRTM... and DAXRT...Y...), Crop3D (also see DXRTA.../DXRTM...), BoxFinder (also see DXRTA.../DXRTM..., DXRT...G... and optionally DAXRT...Y...), ExtractProfile (also see DXRTA.../DXRTM...), FillGaps (also see DXRTA.../DXRTM...), Mesh3D (also see DXRTA.../DXRTM...), ModelFinder3D (also see DXRTA.../DXRTM..., DXRT...G... and optionally DAXRT...Y...), PlanetFit (also see DXRTA.../DXRTM... and DAXRT...Y...), Project3D (also see DXRTA.../DXRTM...), Resample3D (also see DXRTA.../DXRTM...), Subsample3D (also see DXRTA.../DXRTM...), and Volume3D (also see DXRTA.../DXRTM... and DAXRT...Y...) steps.
DXRT0000020000	Both DXRT00000R0000 and DXRT0000030000.
DXRT000000Y000	Aurora Design Assistant 10 Metrology package (step). Also required for AlignPlane, BoxFinder (optional), ModelFinder3D (optional), PlaneFit, and Volume3D steps.
DXRT000000Q000	Aurora Design Assistant 10 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 10 Industrial (and Robot) Communications functionality. Note: Not required for Zebra 4Sight EV6/EV7/XV6 and Zebra Iris GTX bundled with Aurora Design Assistant.
DXRT0000000N00	Aurora Design Assistant 10 Classification package. Enables CNNClassIndex, CNNClassMap, CNNObjectDetect, and DL_OCR steps.
DXRT0000000200	Both DXRT0000000Z00 and DXRT0000000N00.
DXRTM2B2B2B200	All Aurora Design Assistant 10 packages. Enables all Aurora Design Assistant steps.

Software License Hardware ID Key

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.
------------	--

Aurora Design Assistant Runtime License Hardware Key

DXRTxxxxxxxx00U	Pre-programmed Aurora Design Assistant 10 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 [Patents](#) for more information.
 2. Certification pending.
 3. Through an update.