Section 2: Starting an Application

COGNEX

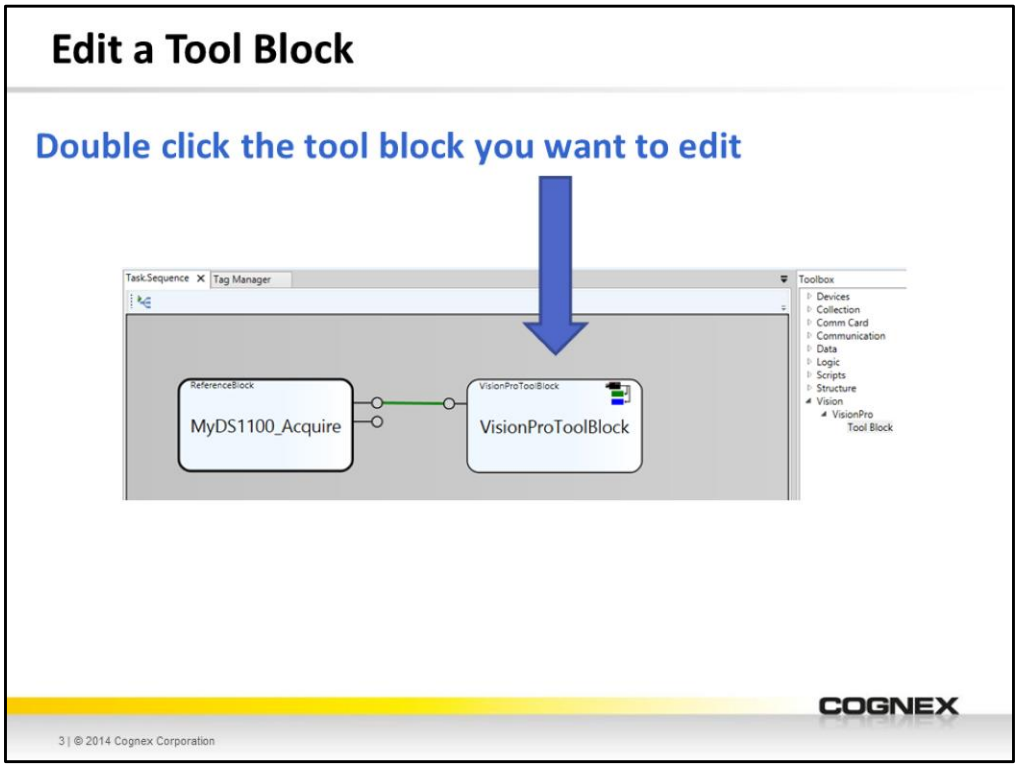
Objective

Process and normalize images

- Fill in missing pixels using a CogIPOneImageTool
- Find the exact location of the part using a PMAlignTool
- Learn Work Area Management

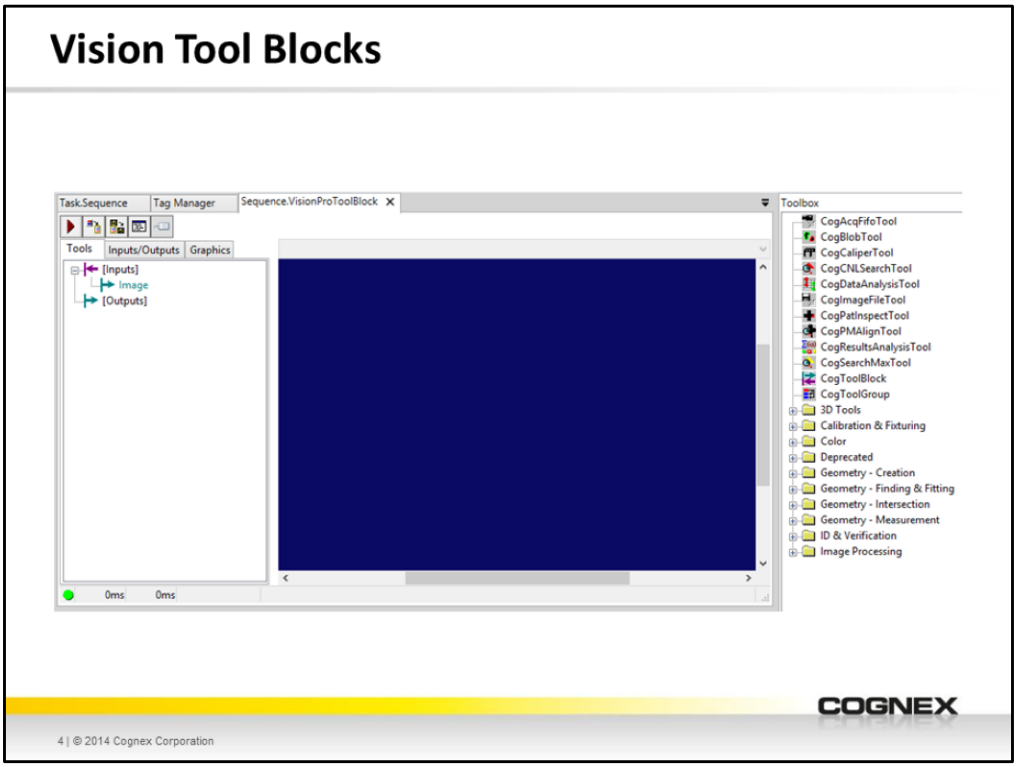
❖ Lab: Add a Vision Tool Block with a CogPMOneImageTool and a CogPMAlignTool

COGNEX



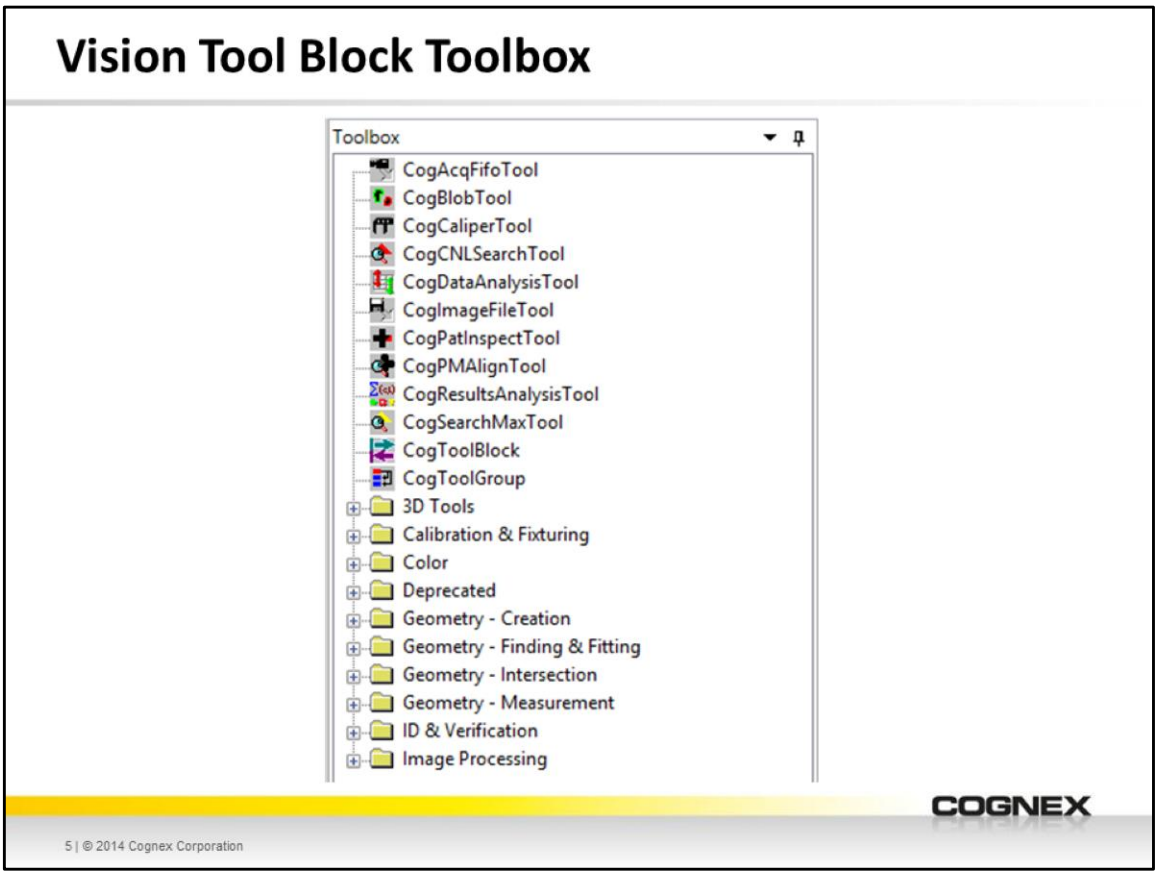
Let's start by adding a ToolBlock to the sequence and executing that sequence.

1) Expand to Vision→VisionPro→Tool Block in the sequence tool box.
2) Drag and drop a Tool Block into your sequence to the right of the acquisition device block.
3) Connect the image of the image block to the new Tool Block.
4) Execute your sequence once to:
    - Capture an image
    - Pass that image from the camera into the new Tool Block
5) Double-click the new Tool Block to open it and look inside.



Double-clicking the Vision Tool Block opens up the interface to allow programming in the Vision Tool Block environment.

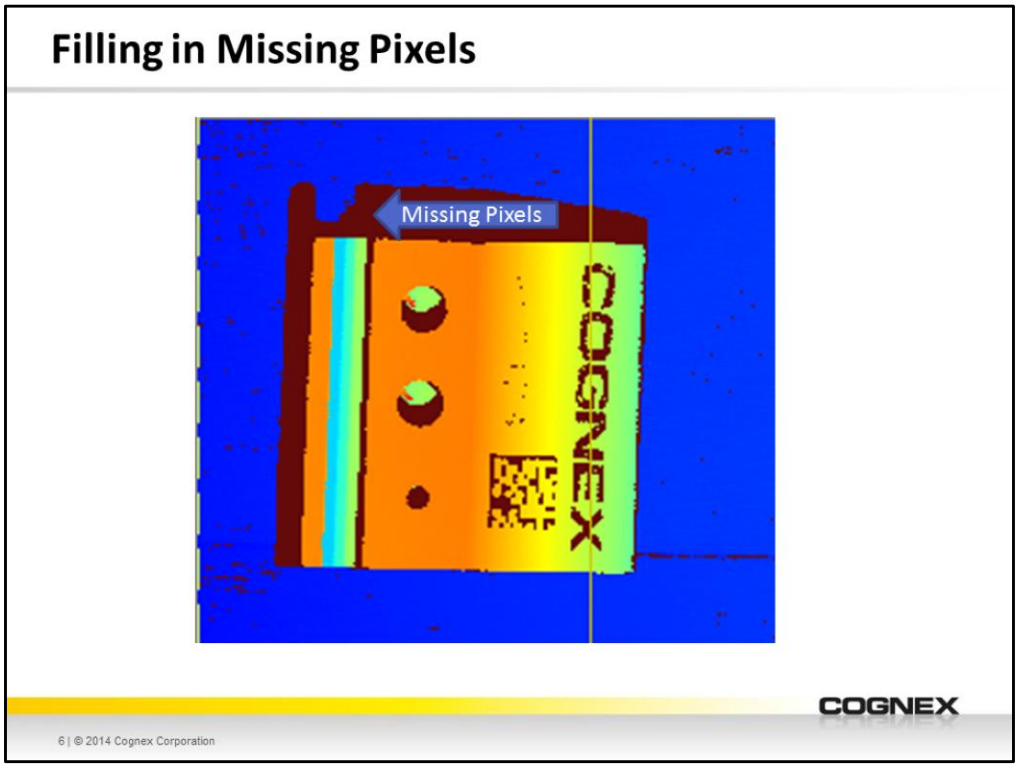
Tools can be dragged from the toolbox on the right to the ToolBlock view in the center.



The Vision Tool Block toolbox will become available to you while viewing the Vision ToolBlock. You can use this toolbox to add vision tools to your tool block.
The following tools and categories help you implement your vision solution:

- CogAcqFifoTool: Creates a custom (provides additional acquisition control and settings) image acquisition tool separate from your initial acquisition settings.
- CogBlobTool: Useful for finding dark objects on a light background (and vice versa).
- CogCaliperTool: Allows edge finding and analysis.
- CogCNLSearchTool: Pixel based modeling and finding tool.
- CogDataAnalysisTool: Enables tolerance checking for tool results.
- CogImageFileTool: Custom recording and playback of image on the hard drive disk (HDD).
- CogPatInspectTool: Advanced defect detection tool; must use CogPMAlignTool.
- CogPMAlignTool: Feature-based pattern recognition search tool. Highest level of positional accuracy available.
- CogResultsAnalysisTool: Data calculation and expression building tool.
- CogSearchMaxTool: Color-enabled pattern training and searching tool.
- CogToolBlock: Allows the creation of a Tool Block within a Tool Block which can contain more tools. Useful for custom behavior or organizing complex tool blocks.
- CogToolGroup: Allows the creation of a Tool Group within a Tool Block which can contain more tools. Useful for custom behavior or organizing complex tool blocks.
- 3D Tools: Tools that focus on extracting and analyzing 3D data from images.
- Calibration & Fixturing: Tools that help calibrate 2D scenes and extract engineering units.
- Color: Tools that focus on extracting and analyzing color image data.
- Depredated: Tools no longer is common use. They have become deprecated due to newer more useful tools. They are still available for backward compatibility in projects.
- Geometry Tools: Tools that help find and compute geometric details from available tool results. For example, points of intersection, right angles, best fit circles.

- ID & Verification: Useful for decoding 2DMatrix codes and recognizing and verifying optical characters (printed human-readable text).
- Image Processing: Tools useful for enhancing images to help reduce noise and or bring out features initially difficult to see.



You may have noticed that because the camera is at a angle from the part, this creates an area where the camera's line of sight is blocked.

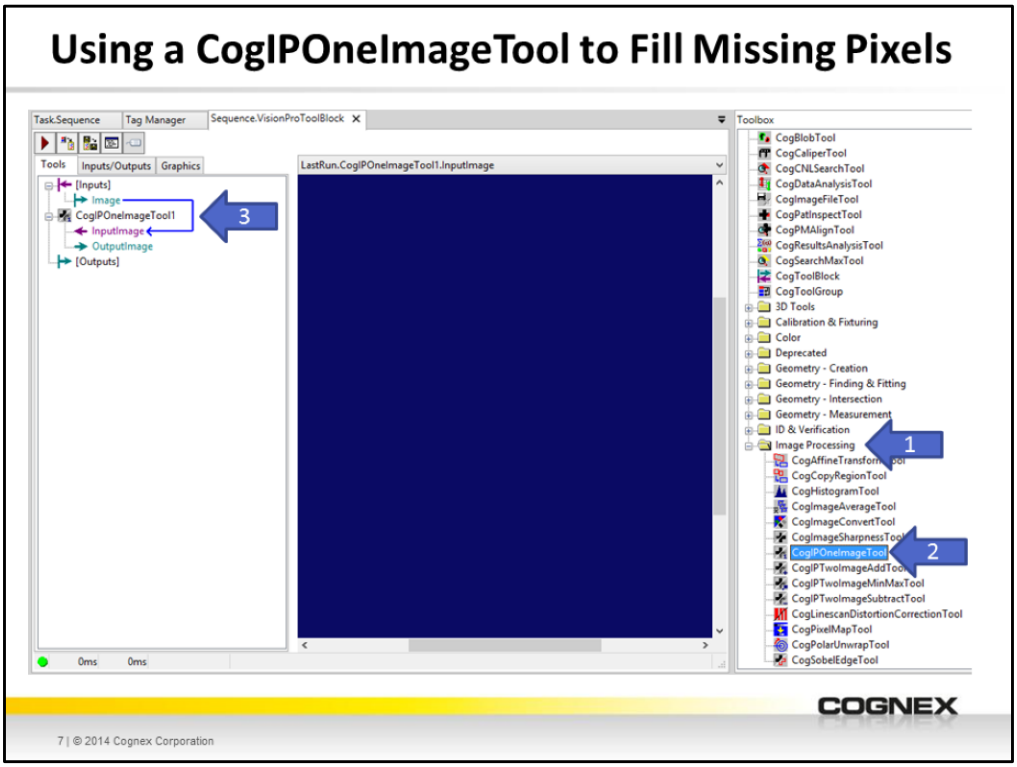
We call this area missing pixels and it is different from all other areas.

This area is not dark or light or high or low like all other areas. This area actually has no pixel data. As opposed to a dark area representing a low point in the 3D space, this missing pixels area has no value to report.

For many purposes, this missing pixels area can be ignores and we can simply focus on areas where we do have data.

Let's explore how to address these areas intelligently so we can:

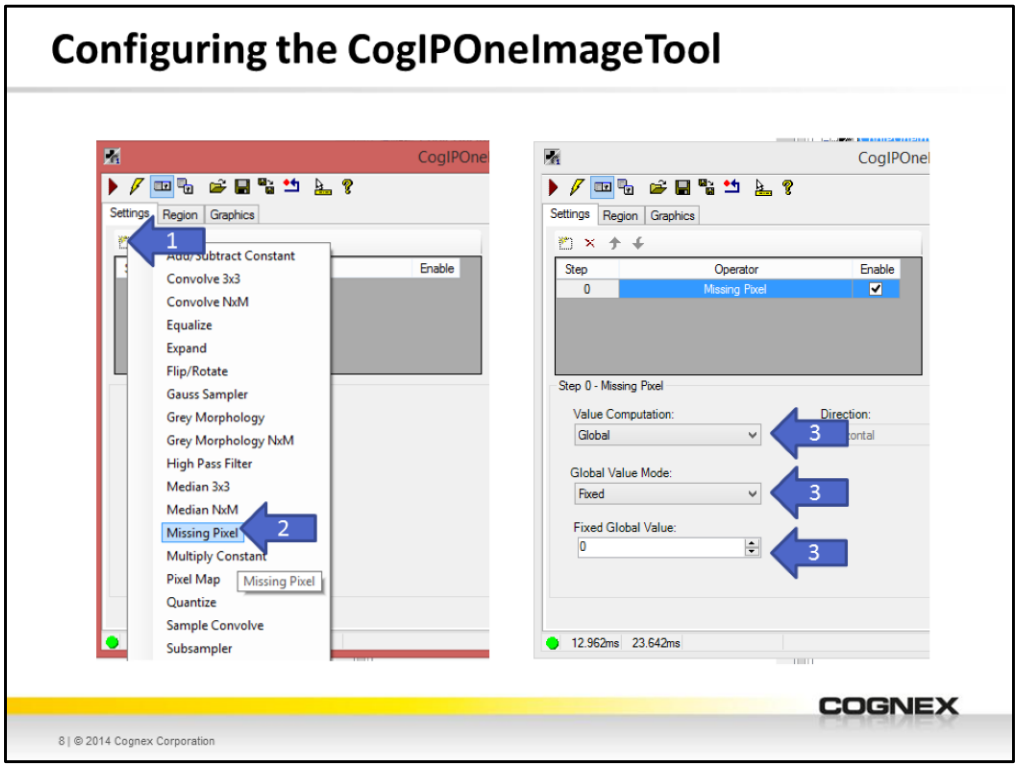
- Avoid potential pitfalls due to lack of data we may run into
- Create an image for display that is more useful when viewing an operator interface



To begin using a CogIPOneImageTool:

1) Expand Image Processing category.
2) Select the CogIPOneImageTool and drag the CogIPOneImageTool into the tool block listing.
3) Connect the image to the InputImage input terminal.

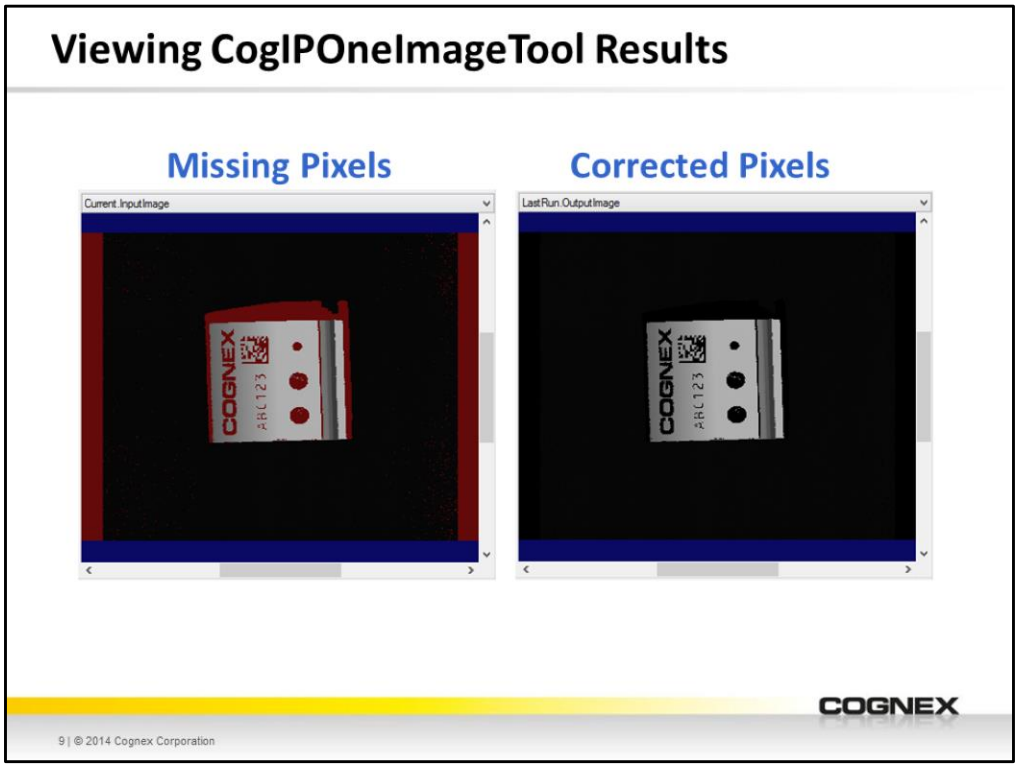
Double-click the new CogIPOneImageTool1 in the tool block listing to open its configuration.



This tool is capable of executing multiple operations on an image.

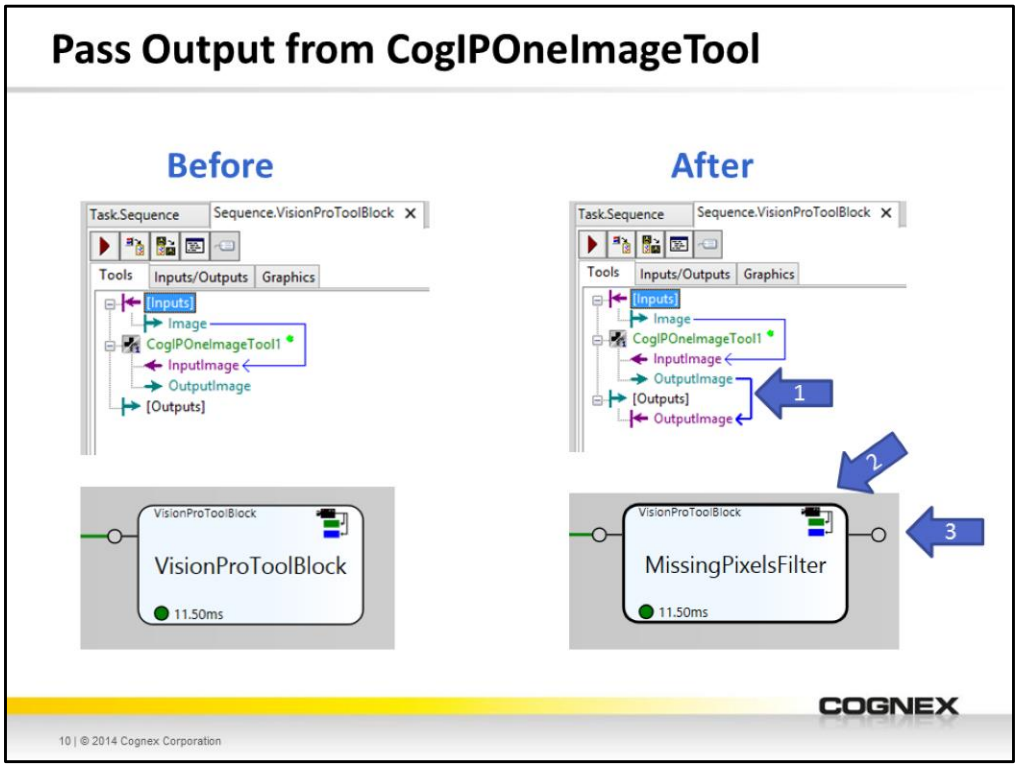
Let's focus on one: Missing Pixels

1) Click the Add New operation button
2) Select Missing Pixels
3) Use the following configuration to make all missing pixels that have no value default to a value of 0
    - Value Computation: Global
    - Global Value Mode: Fixed
    - Fixed Global Value: 0



Use the tool's image drop down menu to view the difference between an image with missing pixels and an image with new pixel values where those missing pixels were originally found.

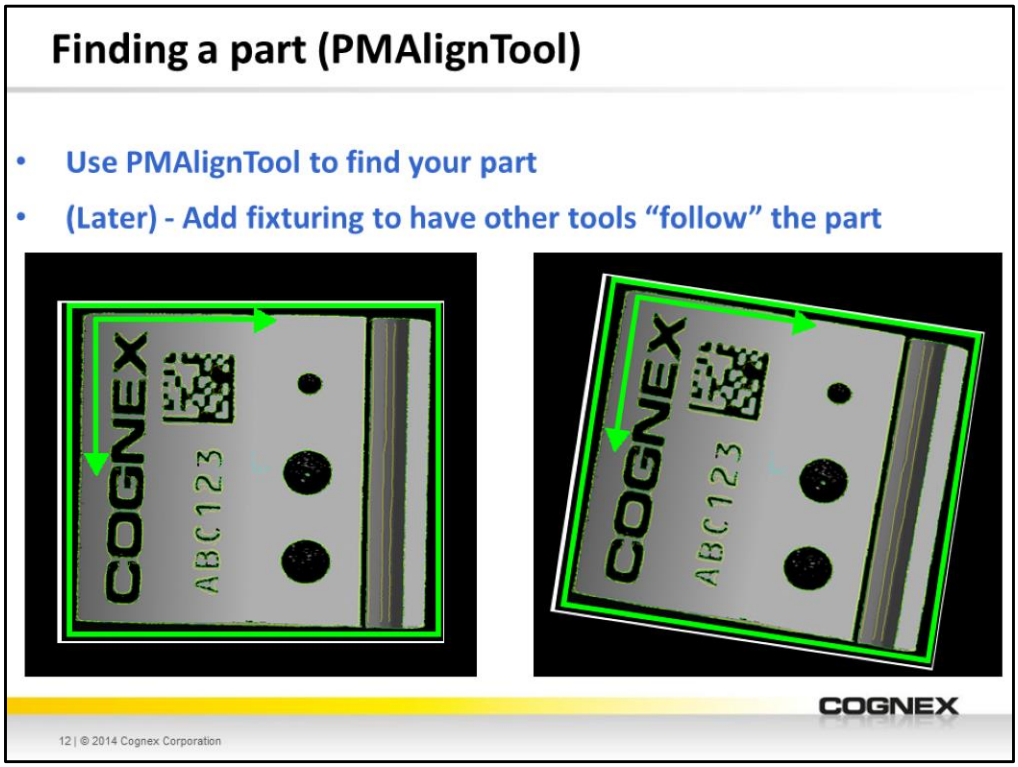
The change is subtle but impactful in terms of the data carried by the image.



Perform the last bit of housekeeping:

1) Drag the OutputImage generated by the CogIPOneImageTool to the [Outputs] collection
2) Rename the Vision Tool Block to MissingPixelsFilter
3) Verify that a new outgoing image is coming out of the right side of the Vision Tool Block

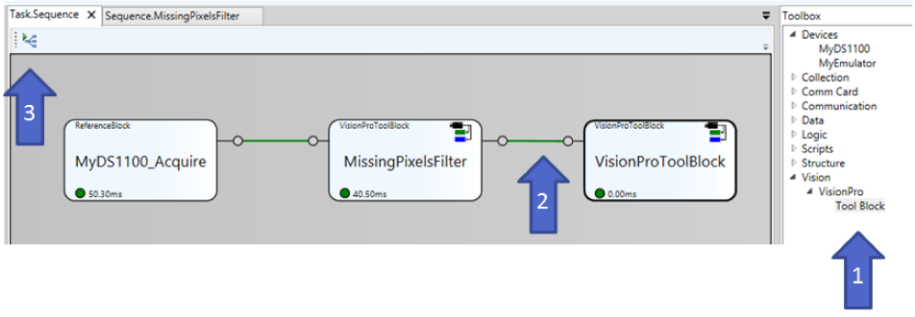
Now since we have set-up the camera to acquire and image, we need to do some inspection on this image as this is a machine vision application. We will add a ToolBlock which will allow us to add VisionPro to the application.



The first thing that needs to be accomplished in the application is the fixturing of the tool. PatMax (PMAlign) will be used as it will find the X/Y location as well as rotation of the trained model in regards to the current image. This information will then be used to adjust the regions of subsequent tools so that their regions follow the part as it moves in the field of view.

Add Another Vision Tool Block

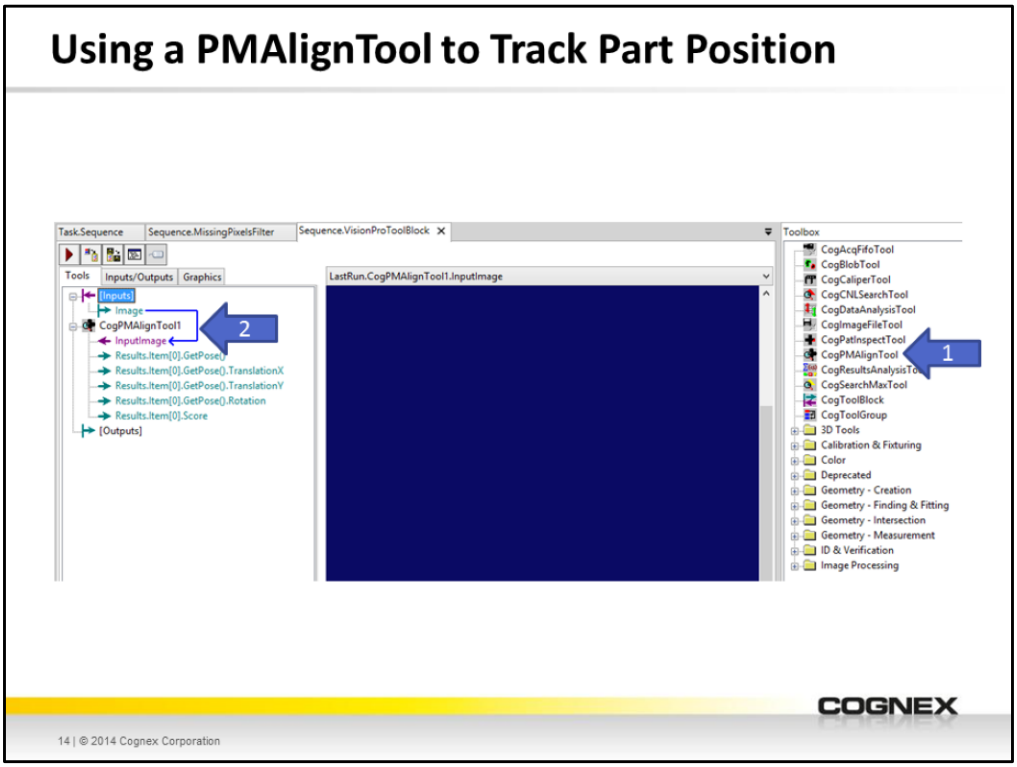
1- Add a VisionPro tool block
2- Connect MissingPixelFilter Output to the new Tool Block
3- Execute your sequence (to move data to the new block)
4- Edit the new tool block



COGNEX

Let's add another Tool Block to the sequence and execute that sequence.

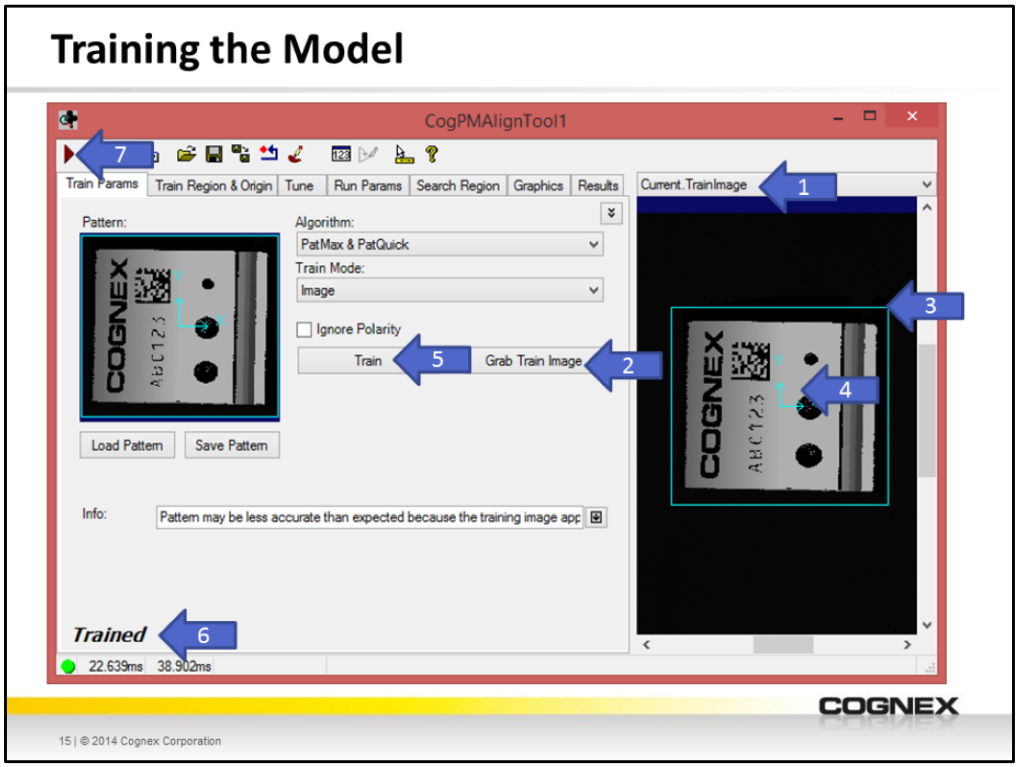
1) Expand to Vision→VisionPro→Tool Block in the sequence tool box.
2) Drag and drop a Tool Block into your sequence to the right of the MissingPixelsFilter block.
3) Connect the image of the MissingPixelsFilter block to the new Tool Block.
4) Execute your sequence once to:
    - Capture an image
    - Pass that image from the camera into the new Tool Block
5) Double-click the new Tool Block to open it and look inside.



To begin using a CogPMAlignTool：

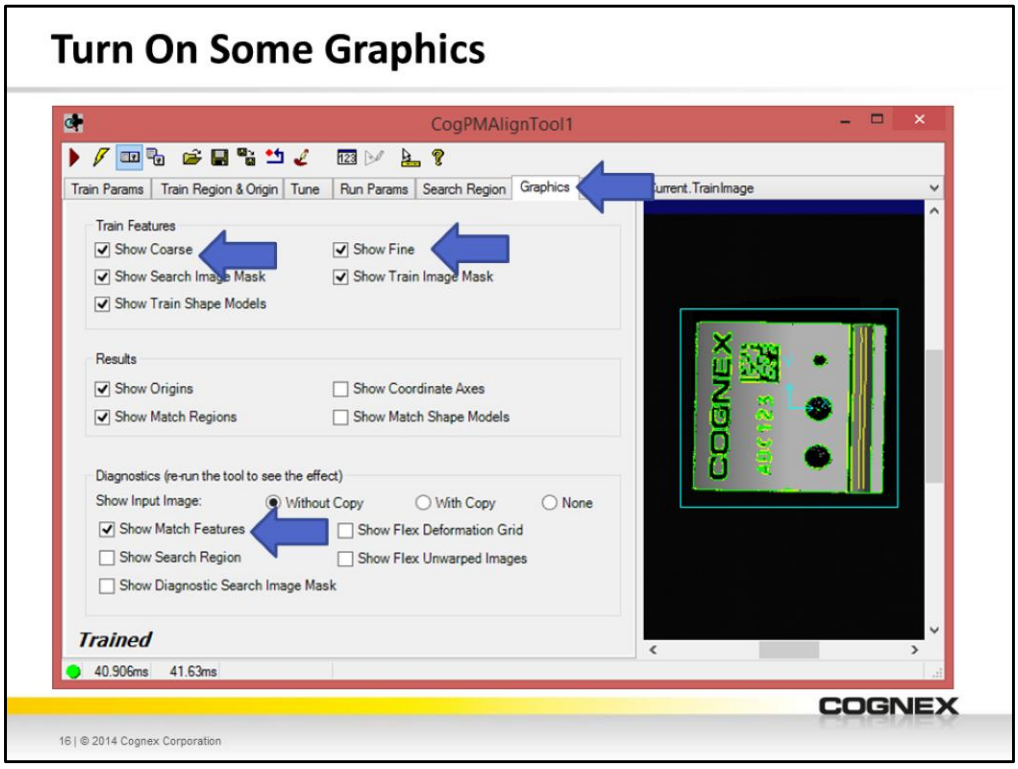
1) Select the CogPMAlignTool and drag the CogPMAlignTool into the tool block listing.
2) Connect the Image to the InputImage input terminal.

Double-click the new CogPMAlignTool in the tool block listing to open its configuration.



Now that you have an image that shows you a good representation of the features you want to train:

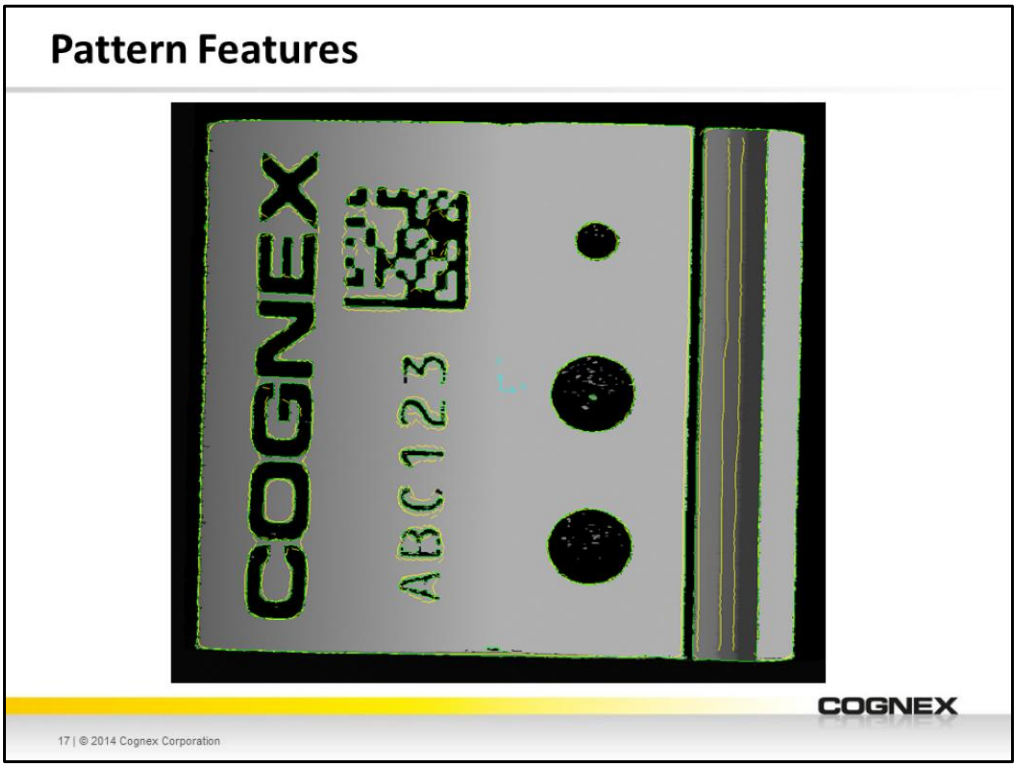
1) Use the tool's image pull-down to select the Current.TrainImage. This will show you the image that is trained. This should be empty since you have never before trained a model.
2) Click the "Grab Train Image" button to bring the Current.InputImage in as the new Current.TrainImage. An image and region graphics should now be visible.
3) Use the region graphics to set the model that you want to train. In this example you can use the entire block, but leep in mind that in many application, you need not train the entire part to make a good model.
4) Drag the origin graphics to the center of the model region.
5) Click the "Train" button.
6) Verify you have successfully trained a model.
7) Click the tool Run button to execute the tool once using the new model and settings.



The Graphics tab allows you to select what graphics you wish to see on the image during training and runtime.

A common setting is to use “Show Match Features” as it will show the features that were trained and found in green while the features that were trained and not found will be shown in red.

Use of graphics does add some time to the program and should often be used as a form of troubleshooting than to be left on at all times.



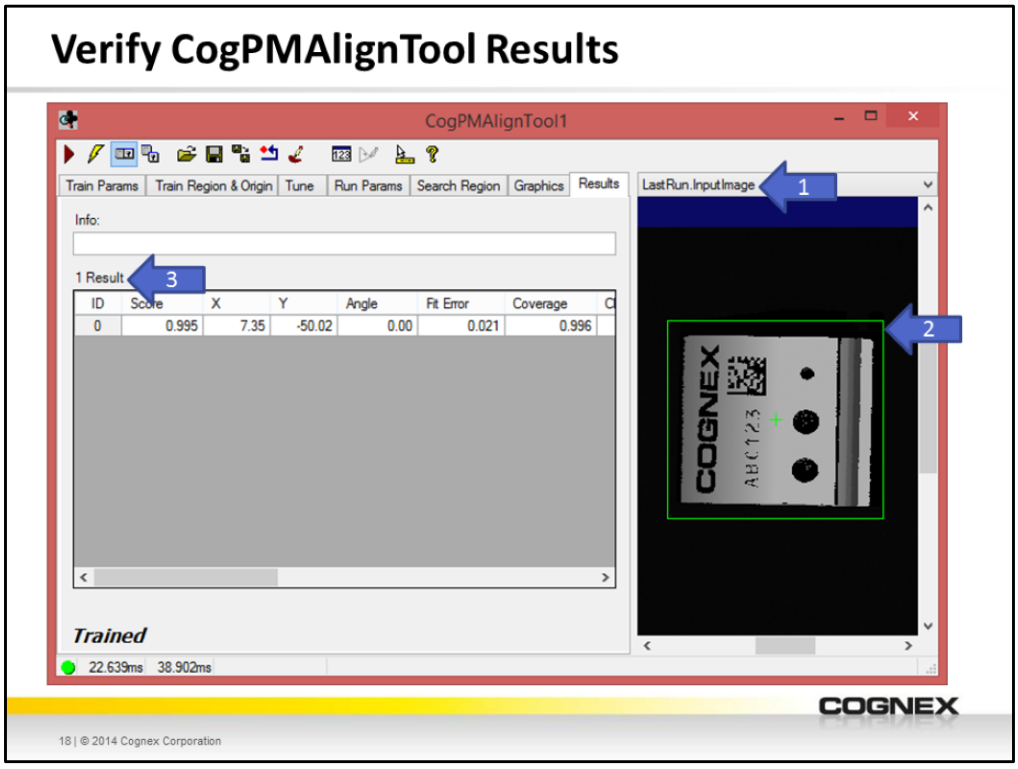
Note that the yellow lines follow the edges of the model rather loosely while the green lines are much tighter to the edges.

This is due to the amount of information that each is using to determine the next feature in the part.

You may need to zoom into the image to see the lines better.

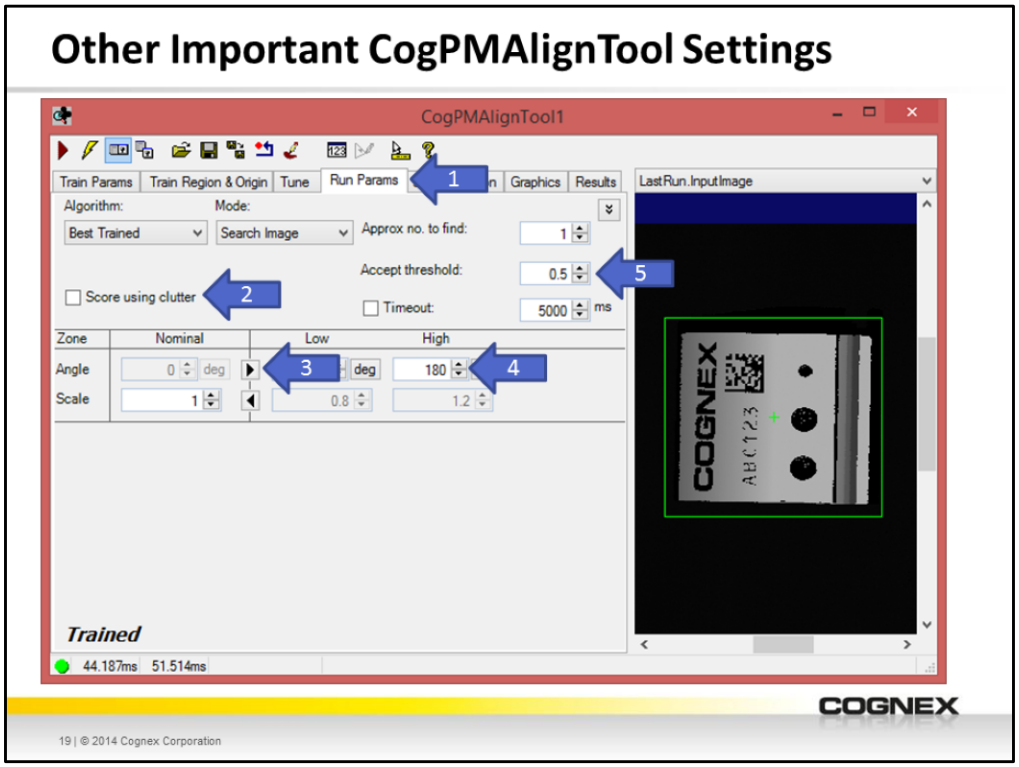
- Yellow lines indicate coarse features; a rough, less detailed extraction of features.

- Green lines indicate fine features; a tight, more detailed extraction of features.



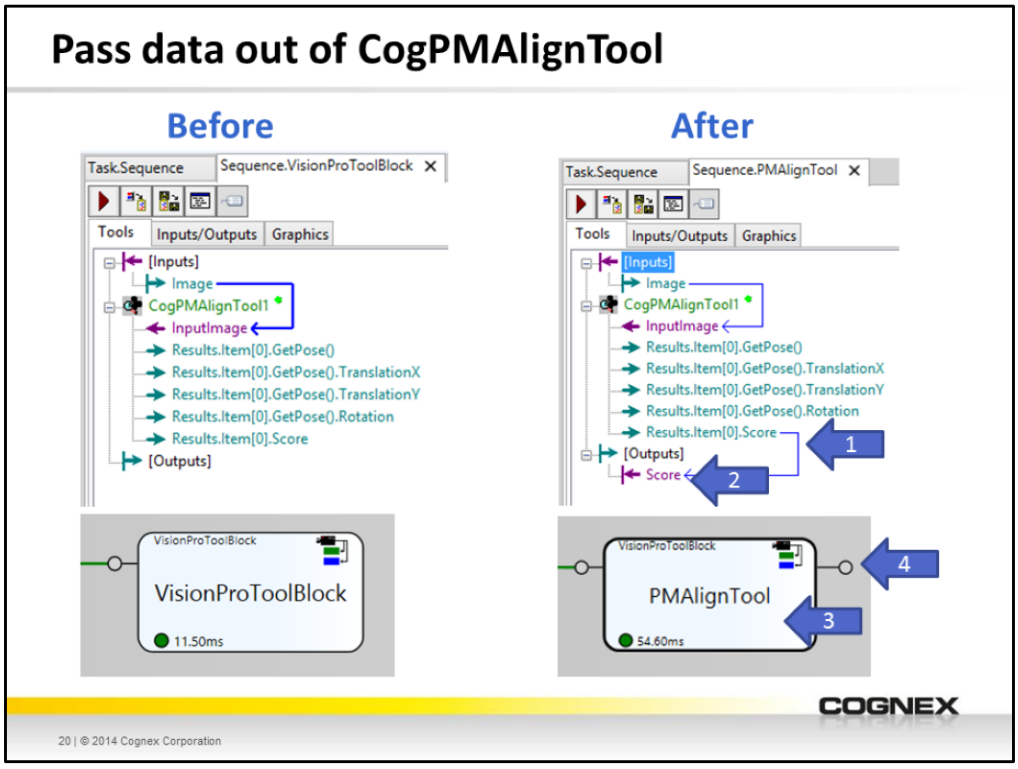
Jump to the "Results" tab and confirm you have found a single matching result.

1) Select the LastRun.InputImage to see the last image the tool executed on.
2) Verify you have result graphics displayed on the image.
3) Verify you have one result reported along with the result's additional details reported:
    - **ID** - Unique zero-based enumeration of the result
    - **Score** - How well the result features match the trained pattern features
    - **X, Y** - The location of the found pattern in terms of the specified origin point
    - **Angle** - The angle of the found pattern relative to the originally trained pattern
        - If nominal angle is used, this always equals the nominal value
    - **Fit Error** (PatMax algorithm only) - The measure of the variance between the shape of the trained pattern and the shape of the pattern instance found in the search image
        - ➢Lower number is better
    - **Coverage** (PatMax algorithm only) - The measure of the extent to which all parts of the trained pattern are also present in the search image
        - ➢Higher number means more trained features found
    - **Clutter** (PatMax algorithm only) – The measure of the extent to which the found object contains features that are not present in the trained pattern
        - ➢Lower the numbers means no "extra" features found
    - **Scale** - The size of the found pattern compared to the originally trained pattern (a.k.a. Uniform Scale)
        - If nominal scale is used, this always equals the nominal value
    - **Scale X, Scale Y** - The size of the found pattern compared to the originally trained pattern in X (horizontal) and Y (vertical) directions
        - If nominal scale is used, this always equals the nominal value



Visit the “Run Params” tab to control the behavior of the tool when it is searching for the model:

1) Go to the Run Params tab
2) Uncheck “Score using clutter” to ask the tool to ignore extra features (noise).
3) In the Angle settings, click the arrow to make it point to the right.
4) Set the Low and High angle ranges to -180 and 180 to allows the part to rotate 360 degrees. If you do this, your part will be found if it rotates.
5) Verify the Accept Threshold setting is set to 0.5. This means that only potentially matching candidates that score above 50% will be reported as a match by the tool.



Perform the last bit of housekeeping:

1) Drag the Results.Item[0].Score generated by the CogPMAlignTool to the [Outputs] collection
2) Rename the default output name to Score
3) Rename the Vision Tool Block to PMAlignTool
4) Verify that a new outgoing score terminal is coming out of the right side of the Vision Tool Block

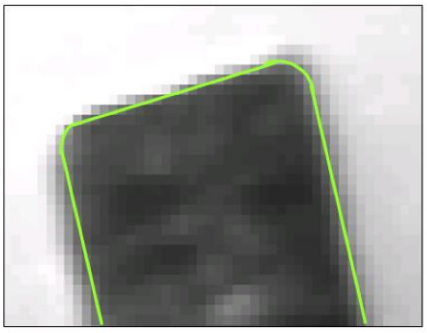
Introducing PatMax®

PatMax® **is a pattern-location search technology**

- **PatMax® patterns are not dependent on the pixel grid**

Returns:

- ***Position*** **of the Pattern**
- ***Size*** **relative to the originally trained pattern**
- ***Angle*** **relative to the originally trained pattern**



COGNEX

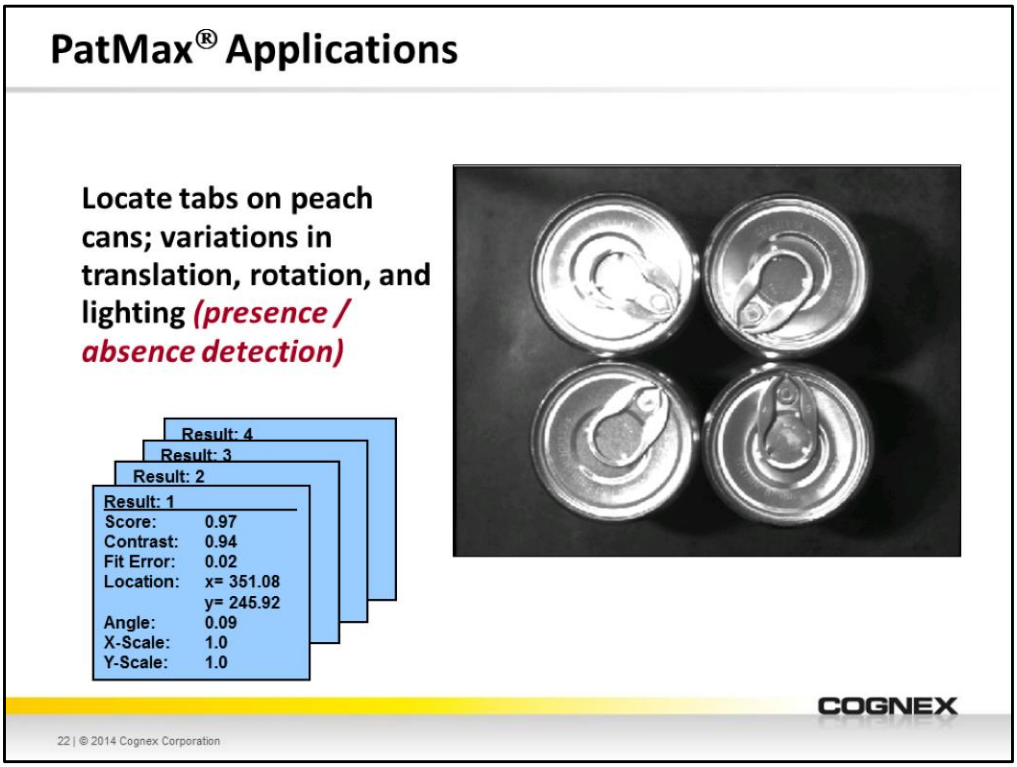
PatMax® is a pattern-location search technology.

A *feature* is a contour that represents the boundary between dissimilar regions in an image.

Feature-based representation can be transformed more quickly and more accurately than pixel-grid representations.

Unprecedented accuracy:

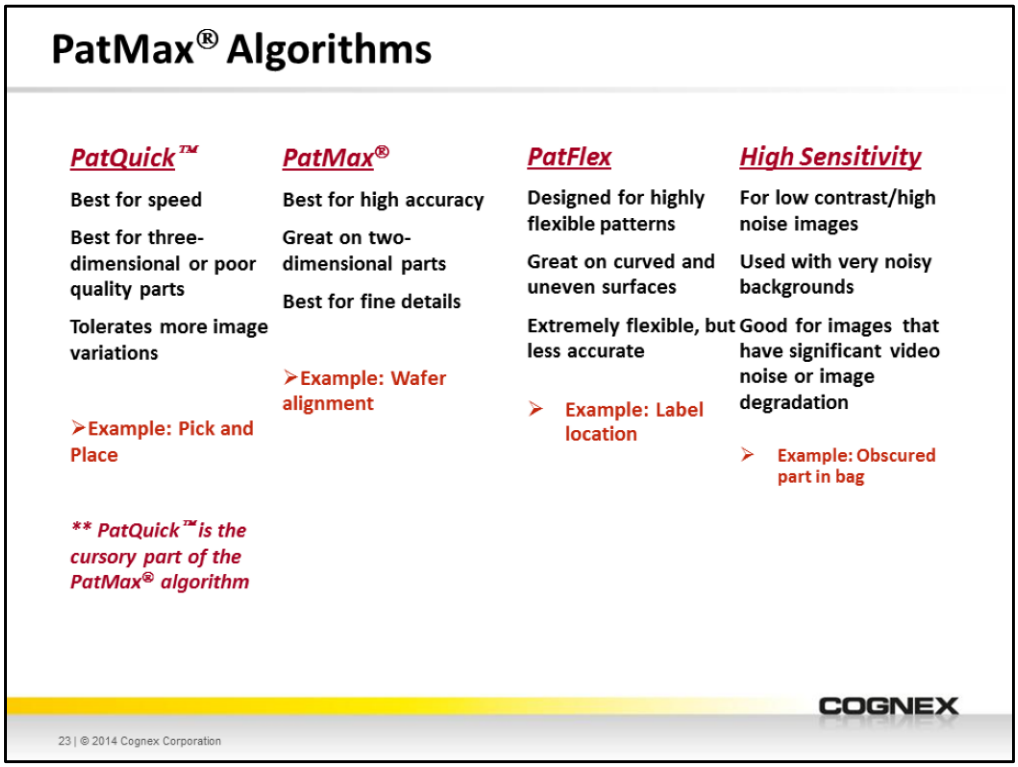
- Up to 1/40 pixel translation
- Up to 1/50 degree rotation
- Up to 0.05% scale



PatMax is not as sensitive to lighting variations.

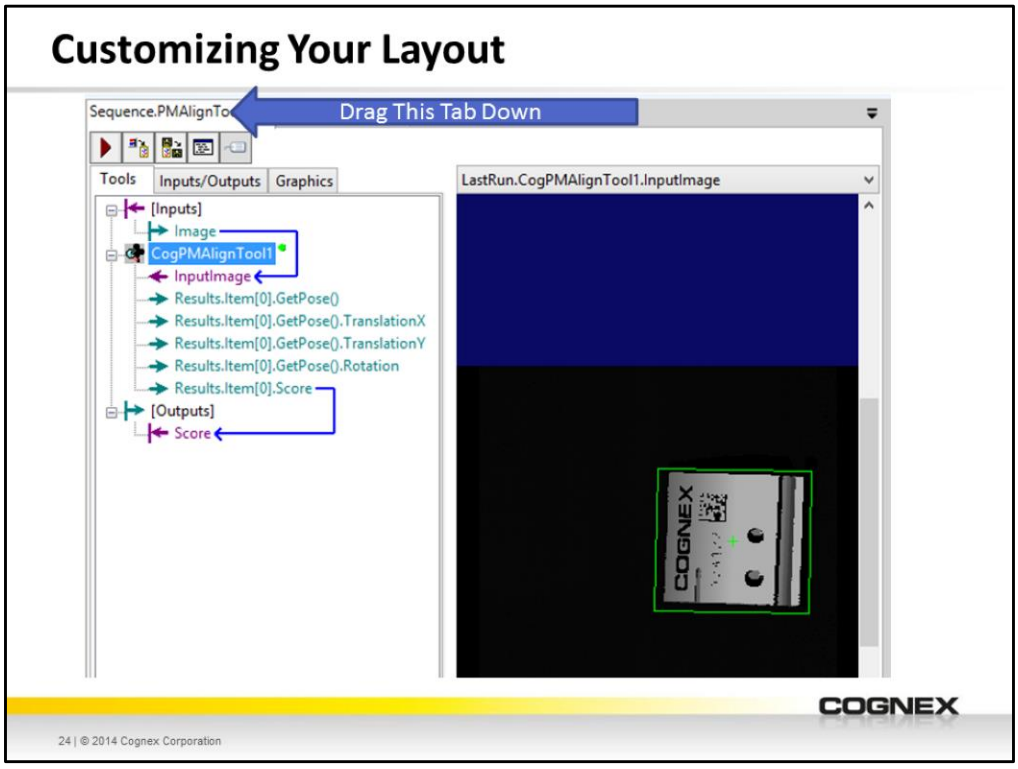
The tabs on the cans above are in different orientations as well as different lighting conditions.

Some tabs look darker while others show glare. PatMax can find them regardless of orientation and lighting effects.



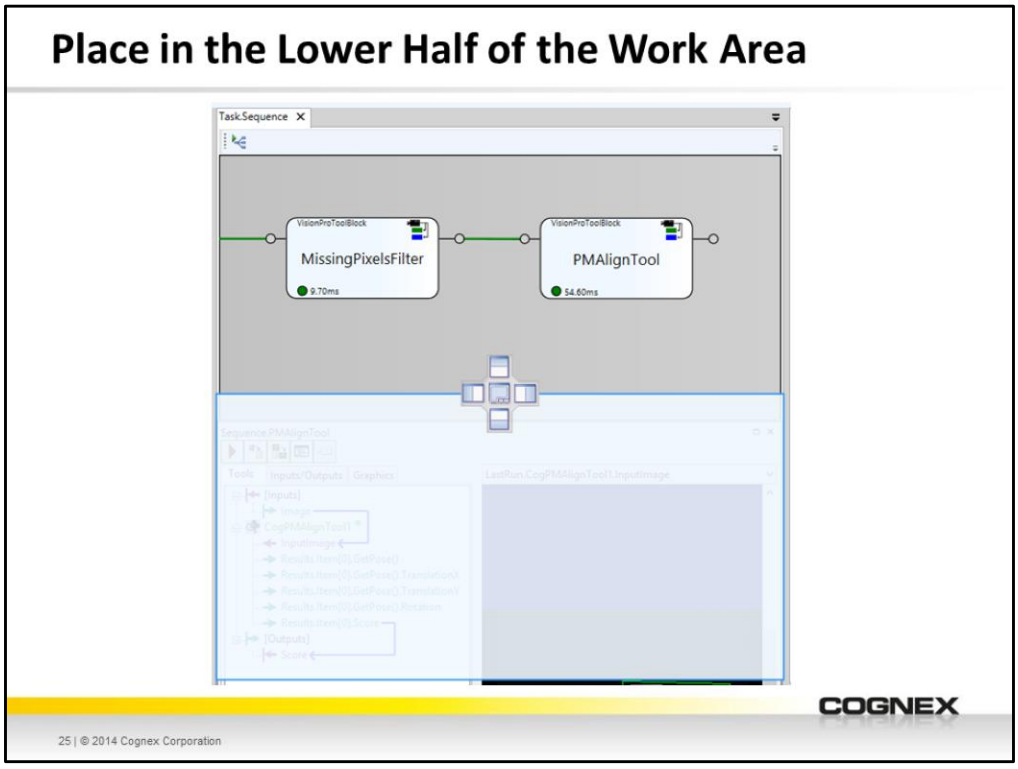
There a few different forms of PatMax that you can activate and use for your application.

- **PatQuick** is the first stage of PatMax. It is a "down and dirty" part finding tool. If speed is needed then this may be a good choice.
    - 3D is stated due to the fact that the part may not stay in focus or have good contrast throughout the 2D acquired image. PatQuick is not as attentive to details.
- **PatMax** will take the potential candidates from PatQuick and perform further analysis. It pays closer attention to the finer details and thus attains the higher accuracy.
- **PatFlex** takes a PatMax model and allows flexibility or movement throughout the image.
    - This can be very useful in label inspections on bags where the 3D nature of the part can tend to hide parts of the model as it is presented in the 2D image.
- **High Sensitivity** is really just a setting for PatMax. It tells PatMax that the image will not have good contrast and may require some extra time and attention to find the object.



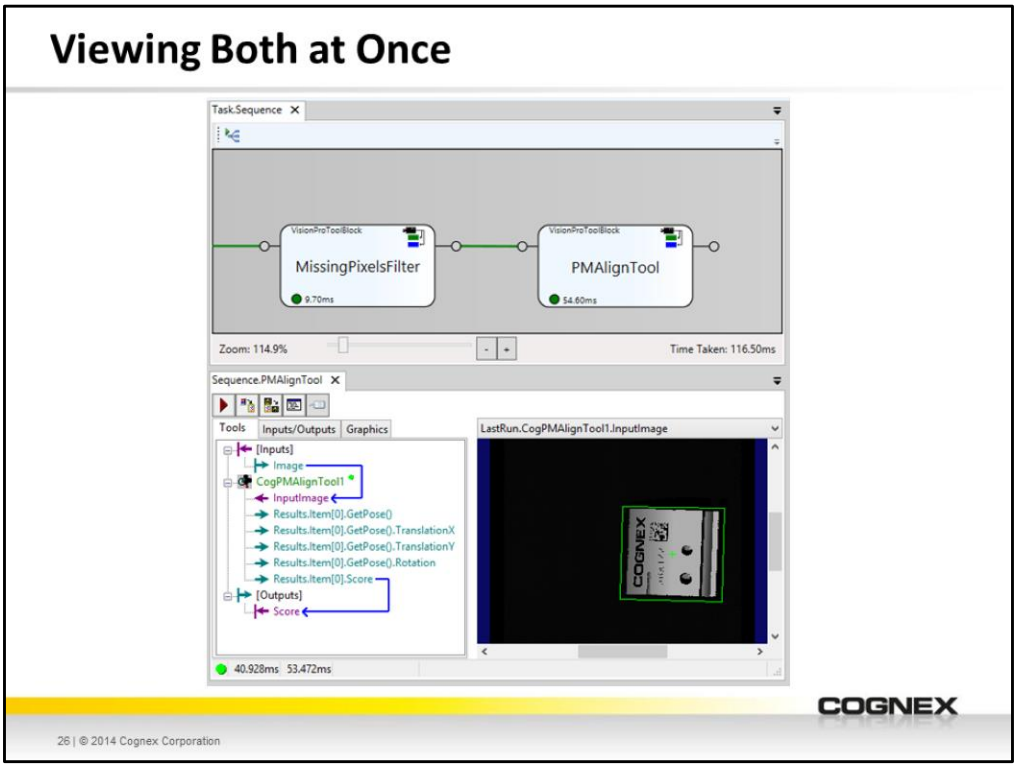
By pulling the tab for the Tool Block down, we are given the docking options for the layout. For now we can place the Tool Block so it shows at the same time as the Sequence.

In this case, I chose the lower placement so I can see the image area if the Tool Block



As you being to drag the tab down, you will be prompted to select another space where you would like it to appear.

Choose the bottom side of the center work area for a new layout configuration.



We will create a better way to display images later in the class with the use of the human machine interface (HMI) also knows as the graphical user interface (GUI) or the operator interface (OI).

For now, you can use this technique to control the sequence and see the tool block executing at the same time.

Summary

Starting an Application

- Expanded and executed the workflow
- Implemented a missing pixels filter
- Implemented PMAlignTool

COGNEX