PROINSPECT

User Manual

Version 2.9

1. Preface

ProInspect is a configurable inspection and robot guidance system. It can be easily tailored for a specific inspection or guidance task through its intuitive user interface.

ProInspect is based on standard PC hardware with the proven vision technology from Cognex Corporation.

This manual contains information about using the ProInspect Vision System. It includes information on the following topics:

- Vision Tools
- Creating recipes
- Installation
- Interfacing ProInspect

The manual assumes the reader to have a basic knowledge of the Microsoft Windows ? operating system.

2. Starting and halting the system

ProInspect in normally set up to start automatically when the system is switched on. The last inspection recipe that was used is activated at start time.

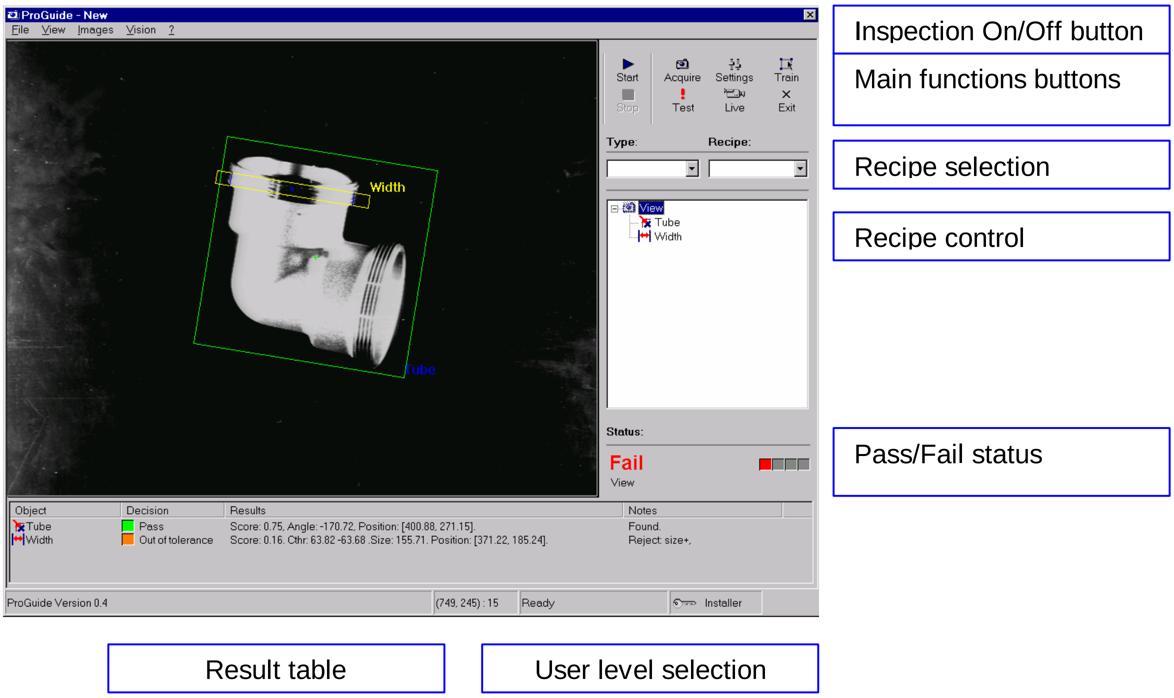
In order to switch off the system, the normal shutdown sequence of Microsoft Windows must be carried out. The sequence can be started by closing the ProInspect application and confirming that you want to switch off the system at the exit dialog.

Important Note: powering off the system without the correct shut down sequence could result in a damaged file storage system.

Your unit may be equipped with a uninterruptible power system (UPS). In this case the shutdown operation is executed automatically and the system can be power off directly.

3. Using ProInspect

Fig. 3-1 ProInspect display



Object	Decision	Results	Notes
Tube	🟩 Pass	Score: 0.75, Angle: -170.72, Position: [400.88, 271.15].	Found.
Width	🟧 Out of tolerance	Score: 0.16. Cthr: 63.82 -63.68 .Size: 155.71. Position: [371.22, 185.24].	Reject: size+,

ProGuide Version 0.4 | (749, 245) : 15 | Ready | Installer

Access levels

In order to simplify and avoid unwanted access to setup operations ProInspect provides three levels of access as depicted in the following table.

Level	Description
Operator	Operator level only allow to alter the display mode, Start and Stop the Inspection and change the active recipe
Engineer	Engineer level allow the user to train and change settings such as sensitivity to defects.
Installer	Full access to all operations including editing inspection recipes.

You can modify the access level by clicking the key icon in the bottom bar or by selecting Access Level from the File menu.

Fig. 3-2 User level control

The system shows a dialog with the three options. You should select the desired access level, enter a password if required and confirm with the OK button.

Note: a password can be required in your particular installation for the Installer and Engineer level.

This manual describes the complete set of operations available at the installer level.

Inspection operations

In normal operations ProInspect is triggered by an external interface to carry out the measurements and inspection tasks defined by the active inspection recipe.

One of several possible interfaces is used to start the inspection sequence:

? PC parallel port
? Serial port
? Ethernet

Other interfaces might be supported for your particular installation. Please refer to the appendices of this manual for details of your specific interface.

The inspection sequence might involve several "views" taken by different cameras and many inspection tasks for each view.

After the inspection is carried out the results are given back through the selected interface and shown on the system display.

The Inspection can be disabled and the system put in a special setup mode. This mode is provided as some configuration and setup tasks cannot be carried out while the inspection is running.

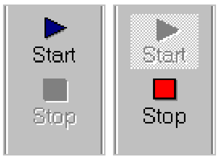
Examples of operations that can only be carried out in setup mode include:

? Changing the active recipe
? Training models
? Creating or editing the recipe

Start / Stop buttons

The Stop and Start buttons allow to stop and restart the automatic inspection mode.

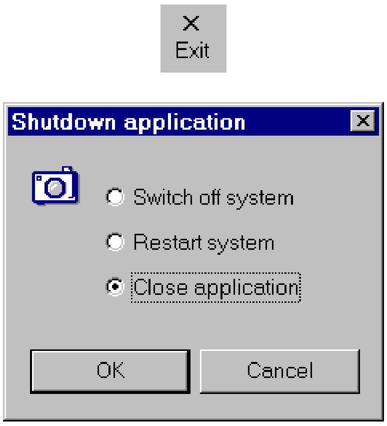
Fig. 3-3 Start / Stop buttons



Exit button

The exit button allows to close ProInspect or shut down the system. Upon clicking the exit button a dialog is shown to confirm the desired operation.

Fig. 3-4 Exit button and dialog



Shutdown application

○ Switch off system

○ Restart system

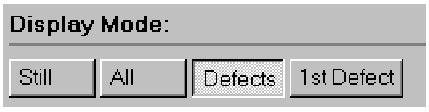
● Close application

Display Mode

ProInspect provides several indications of the inspection results. The results are displayed as a color overlay on the inspected image and individual measurements are shown in the result table. In addition the Pass/Fail decision is indicated in the status area.

The display mode for the graphics overlay and the result table id selected by pressing one of the buttons of the display control as depicted by the following table.

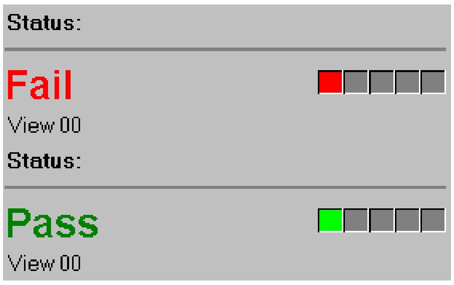
Fig. 3-5 Display mode control



Still	The display on hold
All	Display is updated at any new inspection for the view selected with the view control (see further in this chapter)
Defects	The display is frozen to the image and results of the last reject/failed inspection. The system automatically switches to the view that generated the last defect.
1st Defect	Searches and display the first defective result and switches to the Still mode.

Pass/Fail display

The Pass/Fail decision is shown in the status area. This indication is always updated with the last inspection regardless of the setting of the display mode.



Status:

Fail

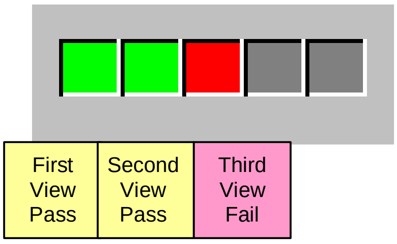
View 00

Status:

Pass

View 00

The green/red square boxes provide an hint on which view caused the reject.



Recipe selection

The recipe selection lets you select the active recipe from the list of recipes that were saved on the disk.

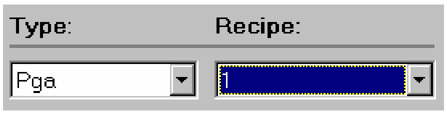
You select two items:

? Recipe: the actual recipe (associated to a file with the same name in the PC disk).

? Type: an optional type name that is associated to a Folder in the PC file system.

The Type is merely a grouping of recipes that helps organizing recipes and maintaining an acceptable length for the recipe list.

Fig. 3-6 Recipe selection



Type: Pga

Recipe: 1

Recipe control

A recipe is the definition of the guidance or inspection task. A recipe consists in a collection of Views and Objects.

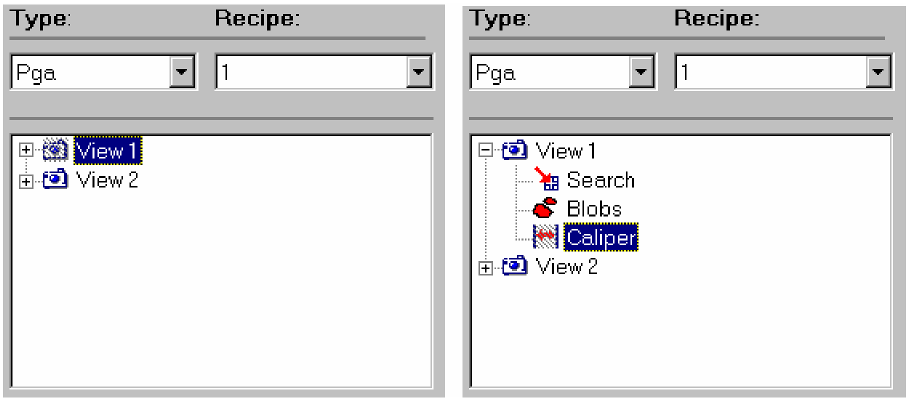
A view is associated to an image taken by one of the cameras connected to the system.

Objects are the definitions of “what to do” for each view.

Recipes, views and objects are described more thoroughly in chapter 4.

The recipe control shows a list of the views in the recipe. The view can be opened to display the hierarchy of objects associated to that particular view.

Fig. 3.7 Recipe control



Type: Pga Recipe: 1

Type: Pga Recipe: 1

Selecting objects

One of the purpose of the recipe control is to select an item of the recipe. You select an object by clicking its icon. A selection has several effects:

? The image associated with the view that contains the selected item is shown in the image display area.

? The results of the objects associated with the current selection are shown in the result table.

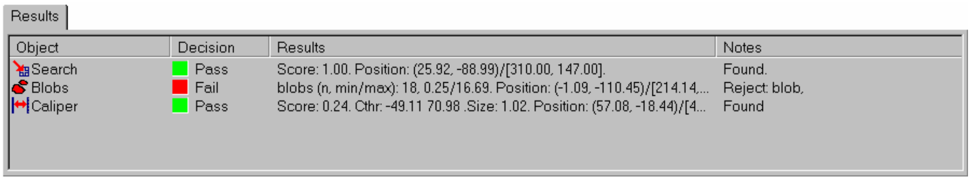
? Certain operations such as training, displaying live images, testing, sampling, calibrating, showing intermediate images are carried out exclusively for view or objects associated with the current selection.

Accessing object's properties

Double clicking an item has the effect of opening a dialog that lets you configure the properties of that particular object. Properties for specific items are described further in this manual.

Result table

Fig. 3-8 Result table



Results

Object	Decision	Results	Notes
Search	🟩 Pass	Score: 1.00. Position: (25.92, -88.99)/[310.00, 147.00].	Found.
Blobs	🟥 Fail	blobs (n, min/max): 18, 0.25/16.69. Position: (-1.09, -110.45)/[214.14,...	Reject: blob,
Caliper	🟩 Pass	Score: 0.24. Cthr: -49.11 70.98 .Size: 1.02. Position: (57.08, -18.44)/[4...	Found

Results for the selected objects are displayed in the result table. For each object the name, the fail/pass status and specific measurement for the type of object are shown as depicted by the above illustration.

If the recipe contains a calibration x,y measurements are shown in a double pixel / units format:

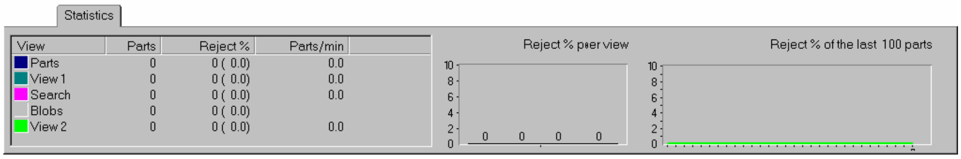
Square brackets such as [127.52, 239,2] are used for pixels.
Round brackets such as (1.31, 2.32) are used for units..

Statistics

ProInspect can provide the following inspection statistics:

? count and percentage of rejects for each view
? count and percentage of rejects for selected objects
? bar graph of rejects per view
? live chart of reject percentages per view.

Fig. 3-9 Statistics page



Statistics

View	Parts	Reject %	Parts/min
🟦 Parts	0	0 (0.0)	0.0
🟦 View 1	0	0 (0.0)	0.0
🟪 Search	0	0 (0.0)	0.0
⬜ Blobs	0	0 (0.0)	
🟩 View 2	0	0 (0.0)	0.0

Acquiring images

Acquire

You can manually acquire an image for test purposes by pressing the acquire button. If the selected view is associated to a camera that requires a “fast trigger” the acquisiton will not take place until the trigger is received. During this period the button will remain pressed. This operation can be aborted by pressing the button a second time.

Live

You can even display a live image by selecting the Live button. Live will carry on until the button is pressed a second time.

Note: several operations cannot be carried out while live display is on. Their controls, buttons or menus, will be disabled until live video is switched off.

Training

Train

Training has two purposes:

? It defines the regions within the image were the analysis takes place.
? It builds reference models that will be used at run time

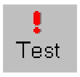
Training involves selecting the relevant regions by dragging a set of cursors over the image of the part to be inspected.

You start training by clicking the Train button. Once all the regions are correctly positioned you end training by clicking it another time

The Golden Image

Training associates positions, sizes and shapes of a set of regions with a reference image, the “Golden Image”. This image is saved with the models in the recipe file and retrieved each time a training session is started.
At times you might need to update the reference image for adjustments of the camera, lighting or for part changes. In order to accomplish that you simply need to click the Acquire button while training.

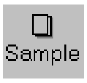
Testing



You can test objects of the recipe by clicking the Test button. You test either an entire view or individual objects depending upon the current selection of the recipe control. The test is carried out with the current image (shown in the image area) . Prior testing you have the option of acquiring a new image or retrieving an image from the disk (Load Image from the Image menu).

The test results are displayed as graphics overlay and shown in the result table.

Statistical Sampling



Some inspection techniques take advantage of statistical sampling. Statistical sampling is a process where a number of images are measured in order to extract useful statistical data. Such data are used to build better models and to automatically define inspection thresholds.

You start sampling by clicking the Sample button. Sampling is carried out for the selected view only. Sampling is completed when the programmed number of images are measured. You can abort sampling by clicking another time the sample button.

When statistical sampling ends successfully models and inspection thresholds are updated. In order to make them non volatile you must save them with the recipe.

4. Recipes

A ProInspect recipe is the definition of the guidance or inspection task. A recipe is made of the entities described in the following section.

Views

Each camera connected to the system provides a view of a part to be located or inspected. They can be multiple views of the same part or views of different parts to be inspected simultaneously such as for a system with multiple inspection stations.

ProInspect currently supports up to 4 cameras or views.

Sites

A Site is a container for vision tools. All the tools in a site may depend upon the position of the site. Their search areas or region of interest are relocated according to the result (x, y, angle) of the locating tool that provides the position of the site. There can be sites without any registration. In this case the site is merely a grouping of tools.

Vision Tools

Vision tools provide a pass/fail decision or some kind of result to other tools. Tools can be combined in a recipe to obtain powerful inspection capabilities. For example the Search tool could find accurately the position of a part before the Caliper tool gauges a feature.

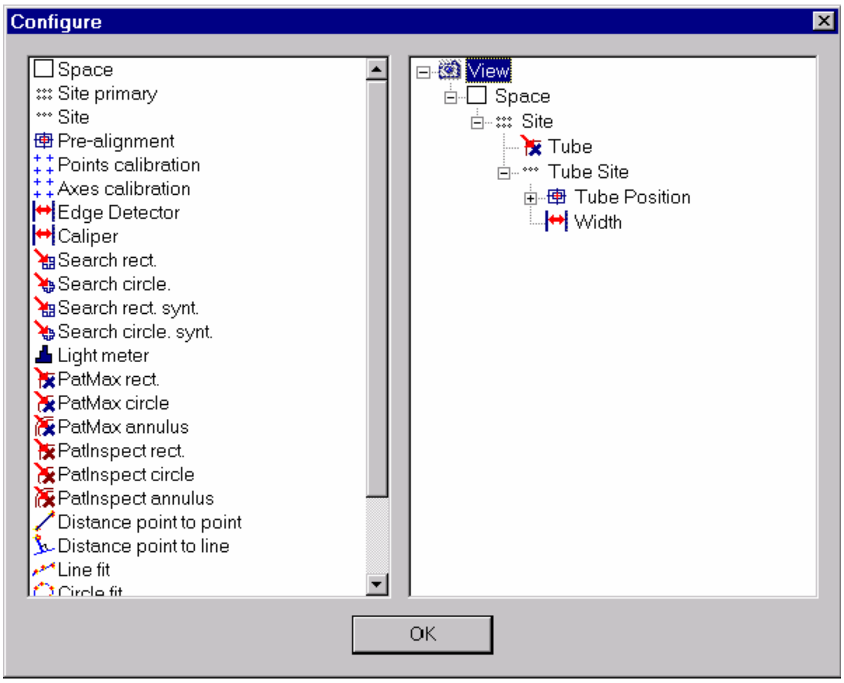


Fig. 4-1 Design / recipe edit dialog

4.1 Recipe settings

Object settings

You can change objects properties through their configuration dialogs accessible by double clicking their icon in the recipe control.

Vision tool properties are described in detail for each tool in chapters 4, 5, 6.

Other object properties

Id's

Each object in the recipe possess an unique Id that is shown in its dialog.

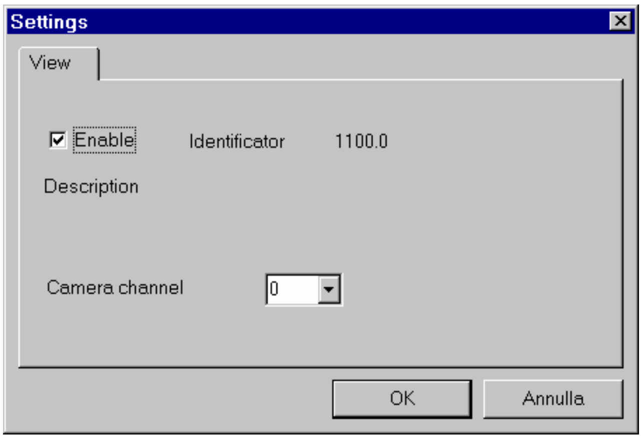
Enable/Disable status

The execution of each object can be enabled or disabled through the Enable checkbox of its dialog.

Camera port (for views)

The view object allows to select the camera port. The current ProInspect version supports up to 4 cameras. More camera properties can be set with the camera properties dialog.

Fig. 4-2 View settings dialog



Settings

View

☑ Enable

Identificator 1100.0

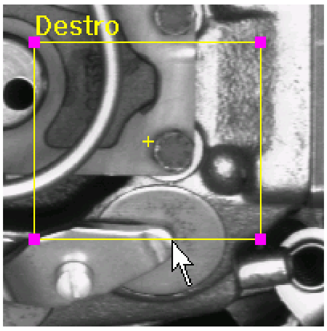
Description

Camera channel 0

4.2 Training

Training establishes the region of interest of inspection tools. Furthermore it acquires image templates for tools that require that. Training is needed after a tool has been inserted in the recipe.

Fig. 4-3 Training a rectangular region



Start Training

Press the Train button or select the Inspect – Train menu to start training.

Select a site

Training is carried out for all the tools of the site selected with the recipe control. If a particular site is selected, you can manipulate only the tools located in that site. Tools that belongs to a space can be manipulated by selecting their space.

Golden Image

Train always recalls the “golden image” of the inspected part. The golden image is stored the first time a recipe is trained. It can be changed simply by acquiring or loading a disk image in a train session.

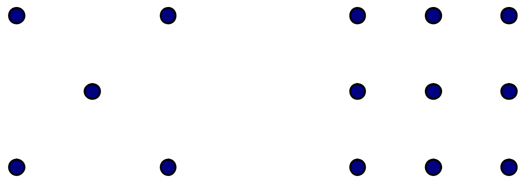
Tool Windows

During training you will relocate and resize the windows that appear as color overlay on the image with the mouse pointer. The active window is shown in yellow with pink handles. In order to resize a window you should drag an handle. Drag the yellow lines to relocate.

Once you are satisfied with the windows positions, press again the train button to conclude training.

4.3 Calibration

Calibration is the process with which the system establishes the relationship between the camera space and a known space. Calibrating involves using known data such as the coordinates of a set of points and measuring them in the image space. Because ProInspect uses error minimizing resolution techniques, the RMS error (root mean squared error) is provided as estimate of the calibration quality.
In designing the calibration experiment, care should be taken to avoid redundancy in the data that may lead to mathematical problems. A typical situation of redundancy is when the calibration points are aligned. Good point configurations are shown in the following figure.



Point calibration

Point calibration works with a small number of points. The acual number of points is established at design time. The World coordinates have to be entered in a table. The Vision coordinates are measured by dragging a set of cursors over the calibration targets.

To start the calibration select a tool in the space you want to calibrate in the recipe tree and activate Vision - Calibrate menu command.

The system shows a set of cursors, one for each calibration point. You should drag the cursors over the targets as accurately as possible. The system shows the camera coordinates in the Pixel X, Pixel Y Columns of the table in the right side of the screen. Enter the corresponding X,Y coordinates and press Done to finish.

Robot calibration

Robot calibration works with a single target that is placed in a set of known position by a manipulator. Prior calibrating a locating tool must be established and trained to locate the target. Robot calibration is done by moving the target to each position. The Robot will issue a specific “Add Calibration Point” command providing the robot coordinates for that point. ProInspect measures the pixel coordinates of the target and adds them to a calibration table. After having completed the process for the desired number of points, the robot issues the “Calibration Solve” command.
This type of calibration offers several significant advantages. The calibration operation is carried out automatically without the need of manual measurements. This calibration maps pixel directly into the robot or machine reference system.

Details of the calibration commands are depicted in appendix A.

Grid calibration

Grid calibration is done by showing a known grid of dots to the system. The system measures the position of each dot and computes the calibration map. A significant advantage of this type of calibration is that the large number of dots enables a better error minimizing mathematics. With this solution it is possible to significantly reduce errors such as the geometrical optical aberration, a particularly significan error with some type of lenses.

4.4 Saving and retrieving recipes

Once you are satisfied with your settings, the recipe should be saved onto the disk. To save a recipe use the File – Save menu command. When you save a recipe the system prompts you to assign a name and a numeric code to the recipe. The numeric code is number from 1 to 255 used for remote selection of the active recipe. To retrieve a recipe from the disk you can use the File – Load Recipe menu command or alternatively select the active recipe directly with the recipe control.

5. Creating a recipe (tutorial)

This chapter describes the process of creating a new recipe. We will use a sample project as a tutorial. We encourage you to try with your system while you read the steps described in this chapter.

Our sample project consists in locating the position and orientation of a pipe fitting on a conveyor and measure a the size one end to check that the part is the correct one.

Creating a blank recipe

We start by creating a blank recipe by selecting New. from the File menu.

Editing the recipe

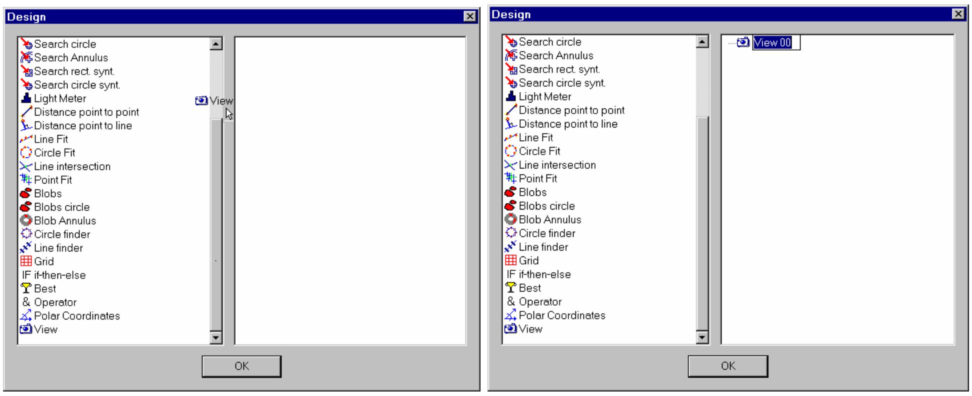
We open the design dialog by selecting Design from the Vision menu. The dialog has two panes. The left pane show the object types that can be added to the recipe. The right pane shows a tree the represents the current structure of the recipe.

Creating a View

We create a view by dragging the camera icon from the left pane to the right pane. Dragging an object from the left to the right pane is a used in general to add an object in the recipe.

After dropping an object one has the option of giving it a name. We called our view “ Conveyor View”. Names can be changed at any time by selecting the object, clicking its name and typing in a new name.

Fig. 5-1 Design dialog

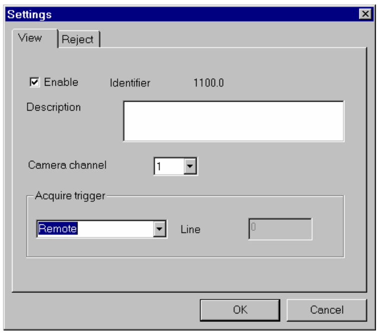


Setting view properties

To set some view properties we open the view dialog by double clicking its icon in the recipe pane.

Object properties can be accessed also outside the design dialog by double clocking their icons in the recipe control.

Fig. 5-2 View settings dialog



☑ Enable

Identifier 1100.0

Description

Camera channel 1

Acquire trigger: Remote

Line 0

We set the trigger property to “ Remote” as the inspection sequence will be started by a command from the robot controlled that is linked ProInspect over a serial line.

We must now test image acquisition. We close the design dialog and clock the Live button. We can now adjust the camera and lighting.

Adding a PatMax Tool

We decided to use PatMax for locating the part as we expect them to have a random orientation. Let’s get back to the design dialog and drag in a PatMax object. We drop the PatMax icon in the view.

Objects in the recipe have a certain hierarchy. Vision tools such as PatMax are always contained in views.

We must set some PatMax parameters.

? Search area is the entire image (default)
? Accept quality is 0.7
? Angle range is –180, + 180.

We do this opening the PatMax dialog by double clicking its icon.

Training the PatMax model

We will now prepare for training. We close the design dialog and press the Train button.

A rectangular cursor is shown in the image area. Once we click near the edges of the rectangle, small squares at each corner appear. Dragging the corners has the effect of resizing the rectangle while dragging the edges moves it.

The small cross in the center of the rectangle is a special point called “ origin” that will be returned as search result. We must rather accurately position the origin.

In order to adjust its position we zoom in by clocking the right mouse button and selecting “Zoom +” .

Once we are satisfied by the position of the model area we press the Train button again to end training.

Testing PatMax

We can now test the PatMax search by moving the part under the camera, acquiring an image and executing the test.

ProInspect finds the part quite accurately. Let move on to the gauging task. We must measure one end of the fitting .

The caliper is a perfect tool for this task. The caliper must be applied with a consistent position and orientation relative to the position of the part.

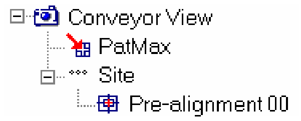
Adding a site

Sites are made for this purpose. Sites get a displacement from the associated set of locating tools. This displacement is applied to the execution of the tools contained in the site.

A site is added to the recipe just like any other object by dragging an icon from the left pane to the recipe view.

Establishing the site position

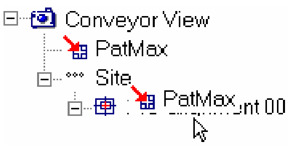
Let’s look at the result in the recipe view:

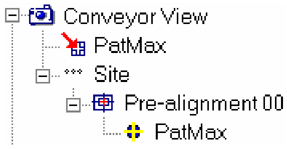


The newly created site has a sub-tree with a Pre-alignment object.

The Pre-alignment is a container for the tool that provides the position to the site.

In order to establish PatMax as the position reference for the site we just drop the PatMax icon into the Pre-alignment.

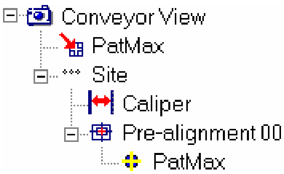




Note that PatMax was dragged from the recipe view and not from the left pane. The instance of PatMax here is merely a reference to an object that exists already in the recipe.

Adding and training a Caliper

We now add the caliper. In order to be correctly positioned the caliper must be contained in the site. We simply drop it into the site.



The caliper like most tools require training. We close the design dialog and start training. We place the caliper region perpendicularly to fitting end that we must measure. The caliper detects the size by locating a pair of edges and measuring their distance,.

We finally have to set some caliper parameters. We do that with the Caliper dialog:

? nominal size = expected size

? size tolerances

? edge polarities light to dark, dark to light.

After testing it we can now save the recipe. We select Save as from the File menu.

6. Vision tool guide

This chapter the introduces ProInspect Vision Tools. Vision Tools analyze the image to extract useful information such as the position or the integrity of an object. Vision tools provide a pass/fail result as well as other measurements such as the position of a part or the size of an object. These data can be communicated to another system or taken into account by other operators. For example a distance tool can consider the position of two points and calculate their distance.

Vision tools are the basis of the powerful inspection capabilities of ProInspect.

This chapter provide a basic description of each tool and depicts the available settings. It includes information on the following tools:

? *Search:* a tool to locate a pattern based on its distribution of gray levels.

? *Edge tool:* a tool to locate an edge within a well-defined area of an image.

? *Caliper:* a tool to locate an edge pair and measure the distance between edges on the edge tool technique.

? *PatMax:* a tool to locate a pattern based on its geometrical properties with outstanding capabilities in locating patterns that are rotated, scaled or incomplete.

? *Light Meter:* a tool that chechs the distribution of gray levels against a trained model.

? *Blob tool:* a tool that finds and counts objects or defects with a gray levels that differ from the background.

? *PatInspect:* A tool that uses PatMax technology to detect and report pattern defects.

For a more through description of the underlying technologies this manual includes a “Vision Tool programmer guide” in appendix A.

6.1 Search

The purpose of Search is to locate and measure the quality of one or more previously trained features in an image.

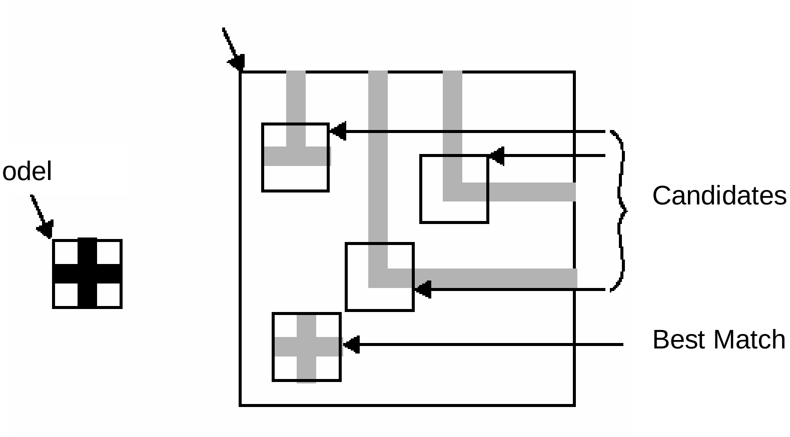
The search operation measures the extent to which a feature in an image matches a previously trained model of that feature.

A feature is any specific pattern in an image. In most cases, you train a representative model from one image and use it to search for similar patterns in that image or in other similar images.

Search locates features by finding the area of the image to which the model is most similar.

Figure 6-1 shows a model and an image, and the areas of the image that are most similar to the model. An image and model similar to those shown in Figure 1 might be used to search for a single instance of a feature such as a fiducial mark on a printed circuit board.

Fig. 6-2 Search candidates



Search Score

Search finds the location of a pattern in a search image based on a model image of that pattern. In addition to returning the location of the pattern in the search image, Search also indicates how closely the pattern in the search image matches the pattern in the model image by returning a score. The score indicates how close a match exists between the trained image and the image whose location was returned. Scores range from 0.0, indicating no similarity between the model and the feature, to 1.0, indicating a perfect match.

Origin point

The location of the pattern is reported for a special point called “origin” . The origin is normally an arbitrary point of the pattern chosen at train time.

Synthetic models

In alternative to models trained from actual images synthetic models are generated by assuming a certain shape of the pattern to be located such as a circle or an annulus. Synthetic models eliminate the need of manual training and provide the additional advantage of having a non arbitrary origin point. For example the origin of a circle is established in the center.

Search tools supported by ProInspect

ProInspect supports the following types of Search tools:

? *Search rect (standard):*	Standard search tool with rectangular model area and rectangular search area.
? *Search circle*	Search tool with circular model area and rectangular search area.
? *Search annulus*	Search tool with annular model area and rectangular search area.
? *Search rect synth.*	Search tool with rectangular synthetic model and rectangular search area.
? *Search circle synth.*	Search tool with circular synthetic model and rectangular search area.

Search Settings

Programmable origin option

When this option is enabled, the model origin can be set graphically at train time. Otherwise the origin is set to the middle point of the model region.

Note: the model origin is an arbitrary point of the model region. The result of the search is the position of this special point in the run time image.

Pattern match limits

Accept - Shape Index	The acceptance level for the search score. Results with scores below this limit are not accepted.
Contrast	A tolerance for the contrast of the pattern found relative to the model. Results with a contrast change higher than this tolerance are not accepted.

Search range

X, Y Search range	The X, Y, in pixels, defines the area around the nominal position of the template in which the system performs the search.

Tolerances

X, Y Tolerances	The tolerances for the search result. The tool fails if the position is found outside these tolerances.
Skip results on edges	An option to avoid considering results that are found along the edges of the search areas as these results are often unreliable.

Number of results

The maximum number of results to search and report..

Search results

Score	Match quality index. A value from 0 to 1 indicating the level of conformity between the model and the pattern found.
Contrast	A measure of the relative contrast of the pattern relative to the model. 1.0 indicates a contrast equal to the trained pattern. Higher or lower values indicate higher or lower contrast.
X, Y	Position of the pattern found.

6.2 Edge tool

The Edge tool offers extremely rapid and precise edge detection and location within a well-defined area of an image.

The Edge tool differs from other vision tools in that it requires that you know the approximate location and characteristics of the feature or edges you want to measure or locate.

The Edge tool is typically used to develop detailed information about the precise locations of features within an object. It is not appropriate for developing information about the shape of objects or features.

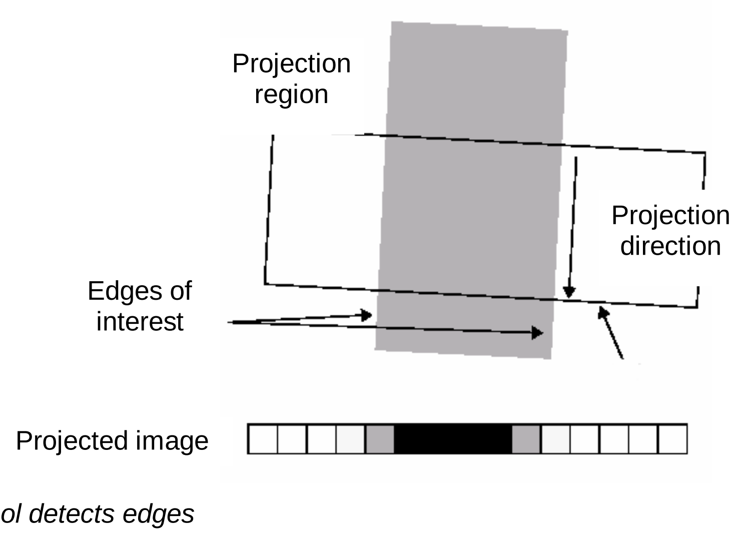
Projections

The first step in using the Edge tool is to specify a projection region within the image to which you want to apply the tool. The Edge tool depends on a carefully defined projection region to isolate just the edge information from a small section of the image.

The projection operation sums all the information in the projection region, accentuating the strength of edges that lie parallel to the projection rays and reducing the effects of noise.

Figure 6-3 shows how a projection can accentuate the edge information in a two-dimensional image. Notice how the strength of the edge is stronger in the projected image than in the initial image.

Fig. 6-4 Accentuating edges strength with projection



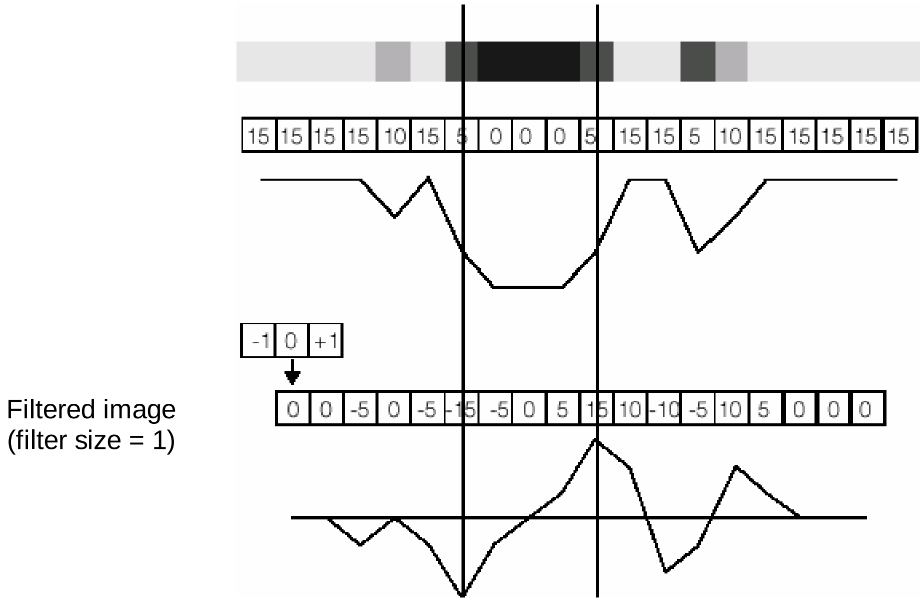
How the Edge tool detects edges

The Edge tool uses the projection region to produce a one-dimensional representation of the portion of the image that contains the edges of interest. This one-dimensional projection image will contain not only the edges of interest, but also other edges caused by noise and unwanted information in the original image. Applying a filter to the one-dimensional projection image increases the strength of the edges of interest while at the same time decreasing image noise.
The filtered image no longer visually resembles the input image. However it does have an important new characteristic: a graph of the filtered pixel values reveals that the peaks in the values, both positive and negative, correspond to the location of the edges within the initial image. The Edge tool uses these

peaks in the filtered image to determine the location of edges in the original image. In addition to producing an image with peaks that correspond to edges in the input image, image filtering also removes noise and spurious edges from the input image.

Figure 6-5 illustrates an example where the image contains two true edges along with spurious edges caused by variations in pixel values around the edge. When a filter with a size of 1 is applied to the image, both the edges of interest and spurious edges appear as peaks in the filtered projection image.

Fig. 6-6 Filtering with filter size of 1



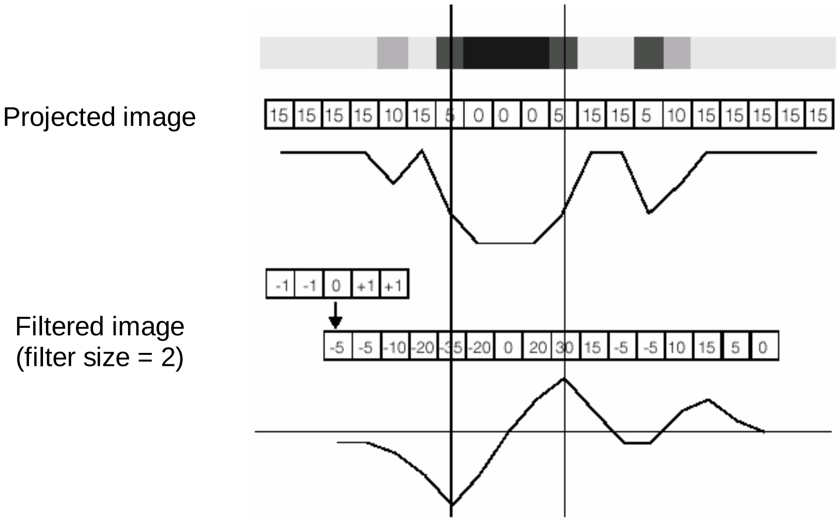
15 15 15 15 10 15 5 0 0 0 5 15 15 5 10 15 15 15 15 15

-1 0 +1

0 0 -5 0 -5 -15 -5 0 5 15 10 -10 -5 10 5 0 0 0

Figure 6-7 shows the effect of applying a filter with a width of 2. The peaks that correspond to the edges of interest are broader, and most of the spurious peaks from Figure 4 are no longer present in the filtered projection image.

Fig. 6-8 Filtering with filter size of 2



15 15 15 15 10 15 5 0 0 0 5 15 15 5 10 15 15 15 15 15

-1 -1 0 +1 +1

-5 -5 -10 -20 -35 -20 0 20 30 15 -5 -5 10 15 5 0

Edge selection

Once it has located the edges in the original image, the Edge tool computes a score for each potential edge pattern in the image. The score for each edge pattern in the image is computed based on a set of selected criteria.

You control the method that the Edge tool uses to score edge pattern candidates by performing the following steps:

- ✍ Select the active criteria's
- ✍ Adjust the weight (importance) of each criterion.

The tool computes a separate value for each attribute for each edge candidate based on the scoring method that you supply, then computes an overall score for each edge pattern candidate by combining the individual scores. It finally selects the edges with the highest score.

Edge tool settings

Tolerances

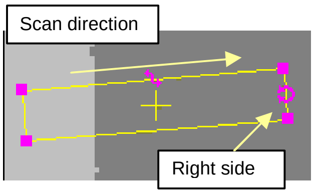
Position offset	The tolerances offset of the edge position found relative to the middle point of the search region established at train time. The tool fails if the offset is outside these tolerances
Accept score	The score acceptance level. The score indicates the conformity of the edge with the specified criteria's. Refer to the "Edge selection" section of this chapter.

Edge detection

Edge polarity	The expected edge polarity. Polarity can be set to one of the following values: ? Light to dark. ? Dark to light ? Don' t care Only edges with the specified polarity are evaluated. All edges are evaluated in "Don't care" mode. The scan direction is relevant in defining the edge polarity. See note et the end of this section.
Contrast threshold	The contrast, in gray levels, above which a transition is considered an edge.
Filter size	The filter width for edge extraction.

Note: The edge tool scan the projection from left to right.The left side is defined at angle 0. The left side can be inverted by rotating the edge tool region. The right is indicated by the rotating handle of the cursor.

Fig. 6-9 Scan direction



Edge selection- Scoring

The checkboxes in this section of the Edge tool dialog let you select which criteria to take into account for calculating the score used for selecting the edge.

Note: The absolute value of the score for a particular criterion depends on the its expected value.

? contrast = 100, exp. contrast = 100, weight = 1 score is 1.0 (perfect edge)
? contrast = 100, expected contrast = 200, weight = 1 score is 0.5

Expected values for contrast are set explicitly in the dialog. The expected value for the position depends on the search region length and is set according to the type of position criterion.

Contrast	Stronger	Stronger edges get higher scores.
	Weaker	Weaker edges get higher scores.
	Custom	Freely programmable scoring function. Please refer to appendix A for details.
Position	Centered	Edges in the center of the search region get higher scores. The score decreases with the distance from the center.
	Closer	Edges closer to the left side of the search region get higher scores. The score decreases with the distance from the left side. Refer to the previous section for a definition of the left side.
	Further	Edges further to the left side of the search region get higher scores. The score increases with the distance from the left side.
	Custom	Freely programmable scoring function. Please refer to appendix A for details.

Edge tool results

Score	Match quality index. A value from 0 to 1 indicating the level of conformity of the edge with the specified criteria's
Contrast	Contrast in gray levels
Position	Position (x, y) of the edge found.

6.3 Caliper

The Caliper tool uses the same techniques described for the edge tool applied to edge pairs. The caliper measures the location and spacing of pairs of edges in images.

The caliper tool differs from the edge tool in that it locates a pair of edges instead of a single edge. By locating two edges it reports as additional result the distance between them.

The following tables depict the settings and outputs of the caliper tool. Most items are identical to the edge tool.

Edge selection - Scoring

Size	Stronger	Stronger edges get higher scores.
	Weaker	Weaker edges get higher scores.
	Custom	Freely programmable scoring function. Please refer to appendix A for details.
Position	Centered	Edges in the center of the search region get higher scores. The score decreases with the distance from the center.
	Closer	Edges closer to the left side of the search region get higher scores. The score decreases with the distance from the left side. Refer to the previous section for a definition of the left side.
	Further	Edges further to the left side of the search region get higher scores. The score increases with the distance from the left side.
	Custom	Freely programmable scoring function. Refer to appendix A for details.

Size – *(Distance between the pair of edges)*	Right	Highest scores are given to edge pairs whose size is closer to the specified " expected value"
	Larger	Highest scores are given to edge pairs with larger size
	Smaller	Highest scores are given to edge pairs with smaller size
	Custom	Freely programmable scoring function. Refer to appendix A for details.

Caliper results

Score	Match quality index. A value from 0 to 1 indicating the level of conformity of the edge with the specified criteria's
Contrast	Contrast in gray levels
Position	Position (x, y) of the edge found.
Size	Distance between the pair of edges

6.4 PatMax

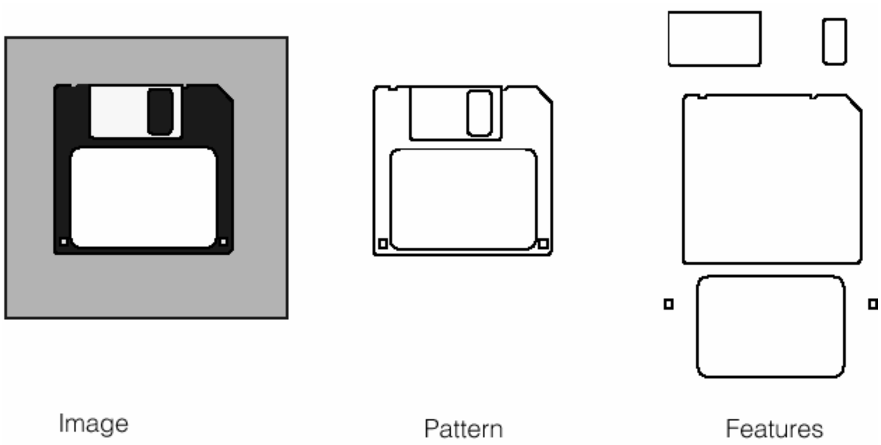
Like other pattern-location technologies, PatMax trains a pattern, then locates one or more instances of that pattern in one or more search images.
PatMax offers three key features that distinguish it from other pattern-location technologies available in machine vision:

? High-speed location of objects whose appearance is rotated, scaled, and/or stretched
? Location technology that is based on object shape, not on grey-scale values
? Very high accuracy

PatMax differs from other pattern-location technologies in that it is not based on pixel grid representations that cannot be efficiently and accurately rotated or scaled. Instead, PatMax uses a feature-based representation that can be transformed quickly and accurately for pattern matching.

Figure 6-10 shows an image, the corresponding pattern, and the features that make up the pattern.

Fig. 6-10 Image, pattern, and features



Note The pattern is made up of both the features and the spatial relationship among the features. Image Pattern Features

Understanding Pattern Transformation

The appearance of an object in an image can vary in several different ways. PatMax can
find objects whose appearance varies in any or all of the following ways:

? Size (overall size change or individual x- and y-axis size change)
? Rotation
? Location

PatMax finds trained patterns in search images no matter what combination of transformations the pattern has undergone. You can limit PatMax to only consider certain degrees of freedom, and within a degree of freedom, only a specified range. By doing this you can ensure that PatMax finds all of the variations that your application encounters in the smallest amount of time.
For each instance of the pattern that PatMax finds in a search image, PatMax returns the location of the instance as well as values for each of the degrees of freedom of the transformation the pattern has undergone.

PatMax also computes a score between 0.0 and 1.0 that provides an indication of how closely the pattern in the search image matches the pattern from the training image after accounting for the transformation that the pattern has undergone.

How PatMax Works

This section provides an overview of how PatMax works. The information in this section will help you understand how to get the most out of PatMax.

PatMax Patterns

When you train a PatMax pattern from an image, PatMax constructs an internal representation of the object. The pattern is a geometric representation of the shapes in the training image. The individual shapes, called features, are made up of continuous boundaries between regions of dissimilar pixels in the image.

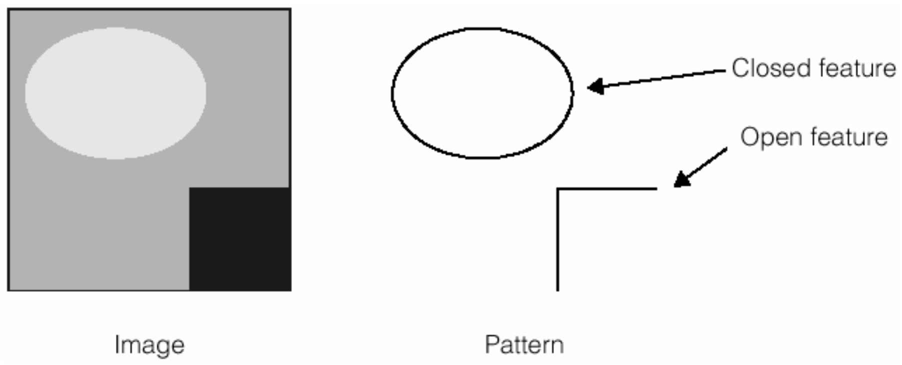
Pattern Features

A PatMax pattern is a collection of features. When you train a pattern from a training image, PatMax isolates all of the features in the image and uses them to train the pattern.

An individual feature is defined to be a continuous boundary between regions of dissimilar pixels. The regions can have different intensity, contrast, or texture. A feature can be open or closed.

Figure. 6-11 shows examples of open and closed features in an image.

Fig. 6-11 Open and closed features

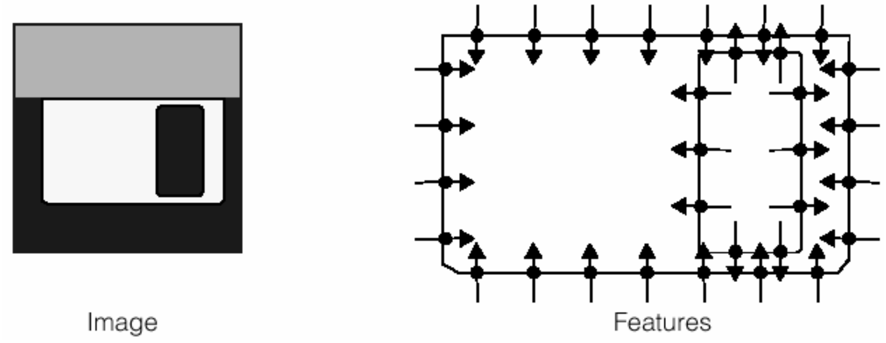


PatMax pattern features are represented by an ordered list of feature boundary points.

A feature boundary point has a location and an angle along with links to its neighboring boundary points. The location of a feature boundary point is a point through which the feature boundary passes. The angle of a feature boundary point is the angle between the image coordinate system x-axis and a line drawn through the feature boundary point perpendicular to the feature boundary and in the dark-to-light direction.

shows the feature boundary points that define a pair of features.

Fig. 6-112 Feature boundary points



Feature Size and Pattern Granularity

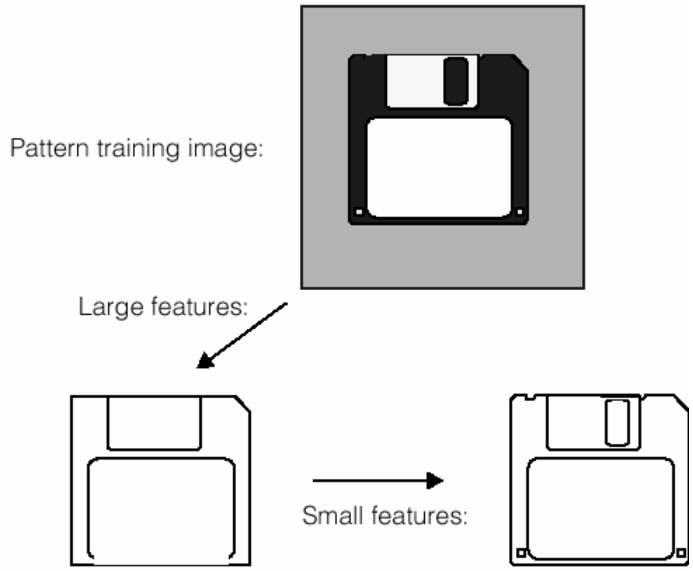
The features that make up a pattern can be of different sizes, from features a few pixels in size to features up to 50 or 100 pixels in size. Most images contain features with a range of sizes.

PatMax uses features of different sizes to locate patterns in images. In general,

PatMax uses large features to locate a pattern in a search image quickly and small features to determine a pattern's location precisely.

Image Features

Fig. 6-12 Large features used for coarse location and small features for fine location

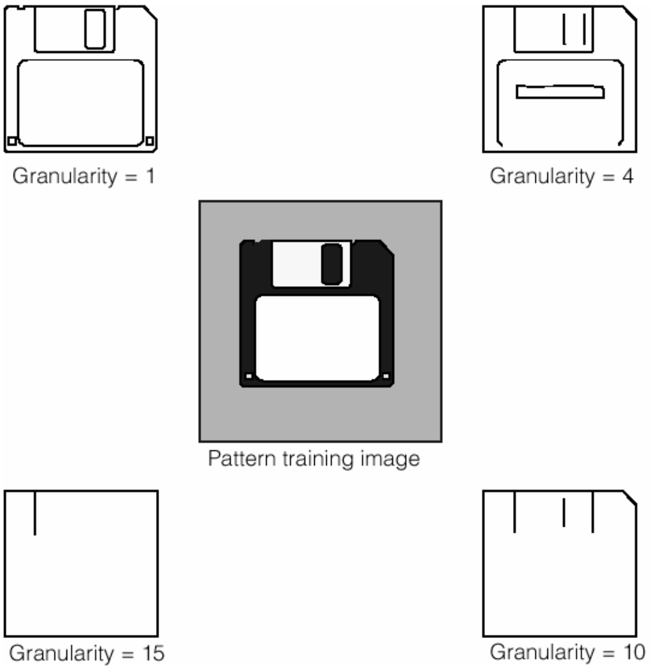


The features that PatMax detects in an image are controlled by the granularity used by the PatMax when it analyzes the image. To detect only the large features in an image, PatMax uses a larger granularity setting. To detect the small features in an image, PatMax uses a smaller granularity.
Granularity is expressed as the radius of interest, in pixels, within which features are detected. Figure 6-13 illustrates two important characteristics of pattern granularity.

? Large features such as the outline of the diskette are detected at both small and large granularity settings.
? Smaller features are present or absent from the image depending on the granularity setting.

In some cases, however, a feature might be present at a fine granularity and at a coarse granularity, but not at an intermediate granularity.

Fig. 6-13 Pattern granularity



At the smallest pattern granularity, the trained pattern includes one or two features for each letter on the diskette label. As the pattern granularity increases, the number of features decreases.
In addition to affecting the features that are trained as part of the pattern, pattern granularity also affects the spacing of boundary points along a feature boundary. In general, the spacing of feature boundary points is approximately equal to the pattern granularity.

PatMax uses a range of pattern granularities when it trains a pattern from an image;

PatMax automatically determines the optimum granularity settings when it trains a pattern. The smallest granularity used to detect features in the pattern training image is called the fine granularity limit. The largest granularity used to detect features in the pattern training image is called the coarse granularity limit.
You can obtain a diagnostic display that shows the actual features and feature boundary points trained using the coarse and fine granularity limits.

Note PatMax trains the pattern using a range of granularities, not just the coarse and fine granularity limits. The coarse and fine limits are the largest and smallest granularities that PatMax uses.

Pattern Polarity

Each of the boundary points that describes a pattern feature has a polarity. The polarity of a boundary point indicates whether the boundary can be characterized as light-to-dark or dark-to-light. You can configure PatMax to find only objects in which every boundary point has the same polarity as the trained pattern, or you can configure PatMax to find objects with mismatched polarity. Ignoring pattern polarity increases the variety of patterns that PatMax finds.

How PatMax Finds Patterns in an Image

This section describes how PatMax finds and reports pattern instances in the search image.

Search Space

When you search for a PatMax pattern in the search image, you define the search space that PatMax uses. The search space is determined by the degrees of freedom you enable, and the range of values to consider within each degree of freedom.

When you run PatMax, it returns a transformation that describes how the trained pattern maps into the found instance. You can use the information in this transformation in two ways:

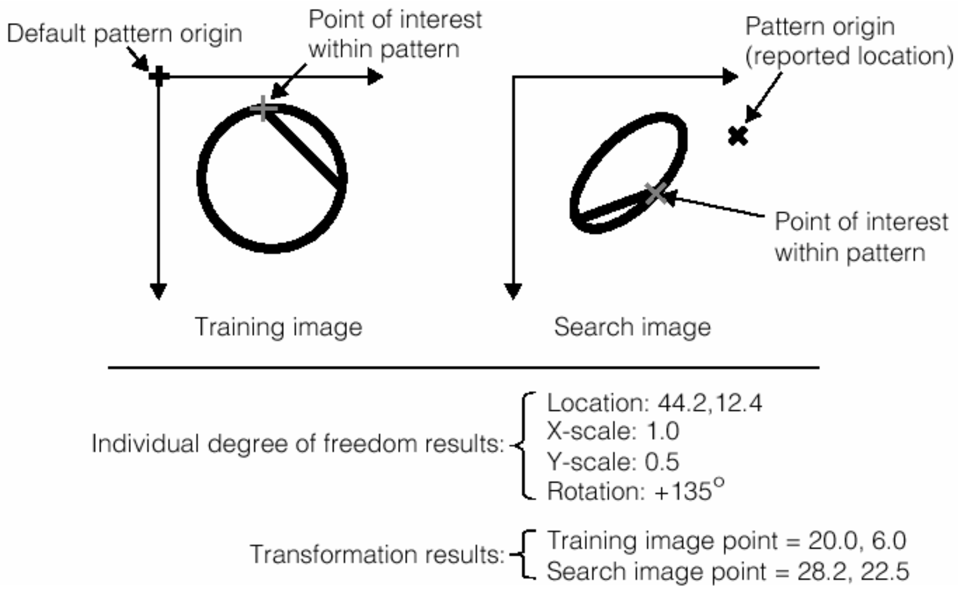
- As a transformation object that you can use to convert any location from the trained pattern to the corresponding location in the search image

- As individual values for the ordinary degrees of freedom (the location of the pattern origin) and individual values for each of the generalized degrees of freedom that you have enabled

When you run PatMax, you can obtain individual results for each enabled degree of freedom, or you can obtain a single transformation object that you can use to transform points between the client coordinate system of the training image (translated by any nonzero pattern origin) and the client coordinate system of the search image.

Figure 6-15 shows an example of a pattern being translated, scaled in the y-axis, and rotated. While you could use the individual values for translation, y-scale, and rotation change to compute where the point of interest is in the search image, simply applying the returned transformation object to the point of interest returns the new point of interest directly.

Fig. 6-14 Using the transformation to locate points of interest


Default pattern origin
Point of interest within pattern
Pattern origin (reported location)
Point of interest within pattern
Training image
Search image
Individual degree of freedom results:
Location: 44.2,12.4
X-scale: 1.0
Y-scale: 0.5
Rotation: +135°
Transformation results:
Training image point = 20.0, 6.0
Search image point = 28.2, 22.5

Score

For each instance of the trained pattern that PatMax finds in the search image, it computes a score value between 0.0 and 1.0. The score an instance receives indicates how closely it matches the trained pattern. A score of 1.0 indicates a perfect match; a score of 0.0 indicates that the pattern does not match at all.

When you specify the Patmax algorithm, PatMax scores instances on both the closeness of pattern fit (the degree to which the shape of the features in the search image conforms to the shape of the features in the trained pattern) and the presence of clutter (extraneous features). When you specify the Patquick algorithm, PatMax scores instances on pattern fit only.

In considering the fit, PatMax considers the shape of the pattern. Differences in brightness or contrast (as long as the polarity is the same) are ignored. (You can specify that PatMax ignore polarity changes in addition to brightness and contrast changes.)

Contrast

In addition to the overall score, PatMax also returns the image contrast of each instance of the pattern it finds in a search image. The contrast is the average difference in grey-level values for all of the boundary points that PatMax matched between the trained pattern and the pattern instance in the search image.

Since PatMax computes the score for a pattern based on the shape of the pattern, the contrast value and score value are generally independent. You can use the contrast value to get additional information about the object.

You can specify a contrast threshold for PatMax searches. If you specify a contrast threshold, only pattern instances where the average difference in grey-level values for all of the boundary points exceeds the contrast threshold are considered by PatMax.

Fit Error, Coverage, and Clutter

PatMax returns three additional score values for each pattern instance it finds in the search image: the fit error, coverage score, and clutter score.

Fit Error

The fit error is a measure of the variance between the shape of the trained pattern and the shape of the pattern instance found in the search image.
The fit error is computed by taking the square root of the sum of the weighted average distances between each boundary point in the pattern and the corresponding boundary point in the pattern instance in the search image. If the pattern instance in the search image is a perfect fit for the trained pattern, the fit error is 0.0. You can use the fit error to assess the degree to which the shape of a pattern instance matches the shape of the trained pattern.

Coverage Score

The coverage score is a measure of the extent to which all parts of the trained pattern are also present in the search image.
The coverage score is computed by determining the proportion of the trained pattern that is found in the search image. If all of the trained pattern is also present in the search image, the coverage score is 1.0. Lower coverage scores indicate that less of the pattern is present.
You can use the coverage score to detect missing or occluded features.

Clutter Score

The clutter score is a measure of the extent to which the found object contains features that are not present in the trained pattern.
The clutter score is the proportion of extraneous features present in the found object relative to the number of features in the trained pattern. A clutter score of 0.0 indicates that the found instance contains no extraneous features. A clutter score of 1.0 indicates that for every feature in the trained pattern there is an additional extraneous feature in the found pattern instance. The clutter score can exceed 1.0.

When PatMax computes the clutter score, it considers all features within the area in the search image that corresponds to the entire image area used to train the pattern.

Controlling the PatMax Search

This section describes how you control the operation of a PatMax search.

Degrees of Freedom

When you perform a pattern search using PatMax, for each generalized degree of freedom (a degree of freedom other than x-translation or y-translation), you must specify either

- That the degree of freedom is enabled or disabled
- If the degree of freedom is enabled you must specify the permitted range of values for that degree of freedom. PatMax will find instances of the pattern that have values for the degree of freedom within the specified zone, and PatMax will compute and report a value for the degree of freedom.

Finally, PatMax might return some model instances that are slightly outside the zone you specify. For example, if you specify a scale between 0.95 and 1.05, PatMax might return results with scale values of 1.09 or 0.91.

.

Elasticity

By default, PatMax requires that each boundary point in the instance of a pattern found in a search image closely correspond to a boundary point in the trained pattern. PatMax can match and identify any change that can be described by a linear geometric transformation (assuming you specify the appropriate degrees of freedom and zones).

When patterns experience nonlinear geometric changes, PatMax can fail to find them, or it can return a low score or inaccurate location information.

You can specify the degree to which you will allow PatMax to tolerate these kinds of elasticity by specifying an elasticity value. You specify the elasticity value in pixels.

In general, you should specify a nonzero elasticity value if you expect inconsistent variation in patterns in search images.

You should keep the following points in mind when specifying a nonzero elasticity value:

- Specifying a nonzero elasticity value does not affect PatMax's execution speed.
- Increasing the elasticity value does not decrease PatMax's accuracy. However, location information returned about additional object instances that are found as a result of increasing the elasticity value can be less accurate.
- If the elasticity value is too low, you will see low scores and your application may fail to find patterns in the search image and/or the positions will be incorrect or unstable.
- If the elasticity value is too high, PatMax may match false instances and may return inaccurate or unstable results.

In general, you should start with an elasticity value of 0; if necessary, increase the value slowly until you obtain satisfactory results.

Ignoring Clutter When Scoring

When you specify the PatMax algorithm, you can tell PatMax to ignore clutter when computing the score of an instance of the pattern in the search image. If you tell PatMax to ignore clutter, then pattern instances receive the same score regardless of the presence of extraneous features.

Using PatMax

This section describes how to use PatMax.

PatMax Training Guidelines

This section provides an overview of the parameters you specify when you train a PatMax model.

Training Image

When you train a PatMax pattern, PatMax uses all of the information in the pattern training image you supply. You should avoid including features in the training image which will not be present in the search image.

In order to train a pattern from a pattern training image, the image must contain distinguishable features. For best results, you should observe the following guidelines when selecting a training image:

- The pattern should include both coarse and fine features.
- The pattern should include information that will let PatMax distinguish instances that vary in all enabled degrees of freedom.

PatMax can return diagnostic information at training time that indicates whether a trained pattern is potentially degenerate in a particular degree of freedom.

Which Features Are Trained

PatMax uses pixels from the root image upon which the pattern training image is based for pattern training. This means that PatMax will train all features found inside the pattern training image window, but it can cause PatMax to detect features outside the pattern training image. The two effects of this change are

- If the pattern training window is close to the edge of the root image, feature boundary points that are closer to the edge of the root image than the coarse granularity limit may not be detected.
If the pattern training window is not close to the edge of the root image then it is guaranteed that every feature inside the window is detected, regardless of the coarse granularity setting.

- PatMax can detect features that are outside of the pattern training image if those features are within the coarse granularity limit of the edge of the pattern training window and if they are not within the coarse granularity limit of the edge of the root image.

PatMax settings – PatMax page

The PatMax page contains general search and tolerances settings.

Programmable origin option

When this option is enabled, the model origin can be set graphically at train time. Otherwise the origin is set to the middle point of the model region.

Note: the model origin is an arbitrary point of the model region. The result of the search is the position of this special point in the run time image.

Search range

Search range mode	Specifies one of the possible settings for defining the search area. ? Entire image ? Centered — Search area centered around the position of the model in the trained image. The size of the search area ? Free — Freely programmable search area. The area is graphically defined at train time.
X, Y Search range	The X, Y, in pixels, defines the area around the nominal position of the template in which the system performs the search.

Tolerances

Position Offset	The tolerances offset of the edge position found relative to the model origin established at train time. The tool fails if the offset is outside these tolerances
Rotation	Rotation relative to the model orientation at train time.

Number of results

The maximum number of results to search and report..

PatMax settings – Alignment page

The alignment page contains settings that influences the search algorithm.

Conformity settings

Elasticity	Degree of tolerance to deformations of the pattern features (contours).
Score	The acceptance level for the global search score. Results with scores below this limit are not accepted.
Contrast	A tolerance for the contrast of the pattern found relative to the model. Results with a contrast change higher than this tolerance are not accepted.
Coverage	Acceptance level for the coverage index. A coverage below this limit is not accepted.
Clutter	Acceptance level for the clutter index. A clutter above this limit is not accepted.
Fit error	Fit error level for the clutter index. A fit error below this limit is not accepted.
Score using clutter	When this option is enabled, the system takes the clutter into account in computing the global score. Else, when the option is disabled, the clutter is non relevant in determining the score.

Tolerances – rotation and scale ranges

Angle	Angle search range
Scale	Scale search range

Pattern granularity limits

Auto-select option	When this option is enable the system perform, at train time, an estimate of the optimal settings for Fine and coarse granularity. In some cases it is interesting avoid autoselecting the granularity values.
Fine	Fine granularity limit
Coarse	Coarse granularity limit

Alignment – Algorithm

Alignment	Specifies what algorithm will be used for searching. ? PatQuick A quicker and less expensive variant of the PatMax algorithm. PatQuick provides the basic PatMax results but lacks separate measurements of clutter, coverage and fit error. PatQuick may yield a lower accuracy's ? PatMax Full PatMax

PatMax results

Score	Match quality index. A value from 0 to 1 indicating the global level of conformity between the model and the pattern found.
Contrast	A measure of the relative contrast of the pattern relative to the model. 1.0 indicates a contrast equal to the trained pattern. Higher or lower values indicate higher or lower contrast.
Fit (error)	The fit error is a measure of the variance between the shape of the trained pattern and the shape of the pattern instance found in the search image.
Coverage	The coverage score is a measure of the extent to which all parts of the trained pattern are also present in the search image
Clutter	The clutter score is a measure of the extent to which the found object contains features that are not present in the trained pattern.
X, Y	Position of the pattern found.
Angle	Angle of the pattern found relative to the model orientation.

6.5 PatInspect

The purpose of PatInspect is to detect defects. PatInspect creates the template image of an object from a sample of training images and then compares run-time images of the object against the template image. For PatInspect, a defect is any change in the run-time image that is beyond the expected variation in the training sample. A defect can be an erroneous or unwanted mark on an object, an incorrectly shaped feature, or the absence of a feature.

PatInspect detects defects according to two distinct modes:

- Intensity difference inspection
- Feature difference inspection

Intensity Difference Inspection

Intensity difference detects differences in grey-scale pixel value between the trained image and a run-time image. Intensity difference includes support for image normalization so that grey-scale pixel value differences caused by lighting variations are not marked as differences.

Feature Difference Inspection

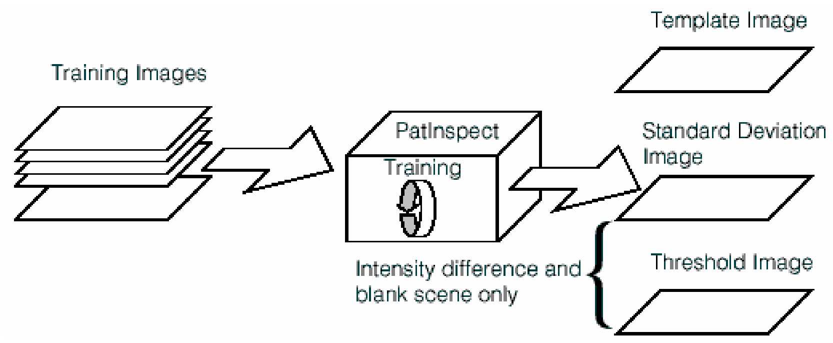
Feature difference detects defects based on difference in feature position, shape, or size between the trained image and the run-time image. Feature difference mode is immune from both uniform and non-uniform intensity differences. Feature difference is also capable of detecting sub-pixel-sized defects.

Figure 5.1 shows an example of the intensity differences and feature differences between two images. The folded corner of the diskette label creates two regions within the image that have different intensity levels. These regions are apparent in the intensity difference image.

Training

During this phase you supply PatInspect with a sample of acceptable instances of the image that you want to inspect at run-time. For intensity difference and blank scene inspection the output of this phase consists of a template image, a standard deviation image and a threshold image. For feature difference inspection the output is just a template image (see Figure 6-16).

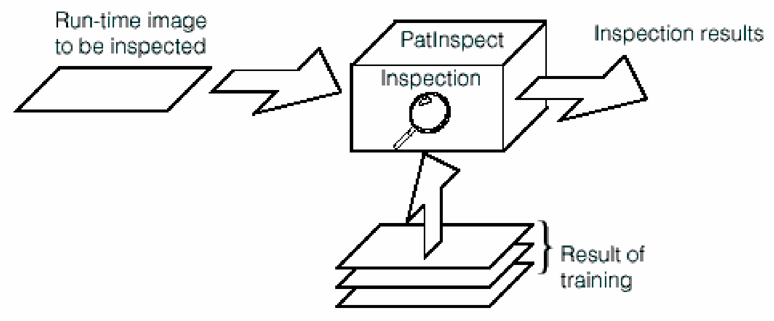
Fig. 6-15 Overview of training



Run-time Inspection

During run-time inspection PatInspect uses the information acquired during training to inspect a run-time image (see Figure 6-17).

Fig. 6-16 Overview of run time inspection



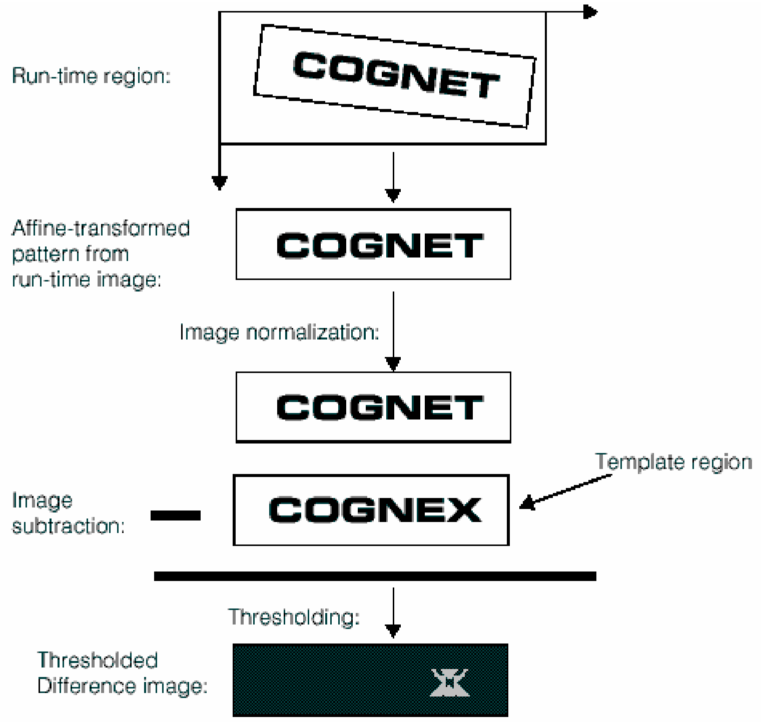
Intensity Difference Inspection

PatInspect intensity difference inspection follows these steps:

1. PatMax locates the alignment region in the run-time image unless you provide an external pose.
2. The run-time region is transformed to be aligned with the corresponding template region.
3. The transformed run-time region is normalized to the template region using a method that you specify.
4. The normalized run-time region is subtracted from the template region to produce a difference region
5. The threshold region is applied to the difference region.
6. The resulting “error image” is segmented into connected defects

Figure 6-18 provides a summary of the steps involved in PatInspect intensity difference inspection. In the example the template region contains the string “COGNEX” while the corresponding run-time inspection region contains the string “COGNET”.

Fig. 6-17 PatInspect Intensity difference inspection



Inspection results and decision

The final result of PatInspect is the count of defects and the area for each defect.

Defects are distinguished in “major” and “minor” defects based on their size. Major defects have a size above a major limit. Minor defects have a size below the major limit and above a minor limit. Defects below the minor limit are neglected.

The pass/fail decision is than taken as described by the following table:

Major defect count > 0	Fail
No major defect, Minor defect count > limit	Fail
No major defect, minor defect count < limit	Pass

PatInspect settings

PatInspect has two type of settings.

? alignment parameters that are identical to the PatInspect tool.
? Inspection parameters\

Refer to the PatInspect chapter for a description of the alignment parameters.

Inspection parameters are described in the following table.

Intensity Inspection settings

Threshold – Sensitivity to defect intensity	Sensitivity index for the intensity of defects. An higher index corresponds to higher sensitivity. The intensity of a defective pixel is a difference in gray levels between the model and the inspected pattern.
Dimension	Sensitivity index for the size of defects. An higher index corresponds to higher sensitivity.
Max number of (defect) areas	Limit for the count of minor defects.
Major defect area	Limit for the size of defect. Defect larger than this limit are deemed as " major defect"

Boundaries inspection settings

Not enabled in this software version

Normalization

Normlization mode	? None / Identity	No normalization is carried out
	? Histogram	Derive the pixel map from the histogram calculated during training. A histogram of the run-time region is generated and compared to the histogram calculated during training. The mapping is constructed so that the histograms match. (That is, at any bin in the histogram of the run-time region, the total number of pixels with a value less than or equal to that bin number equals the total number of pixels in the training histogram with such a value, within a round-off tolerance).
	? Mean and std deviation	Derive the pixel map from the mean and standard deviation of the pixel grey-scale values calculated during training. The mapping is constructed so that the mean and standard deviation of the normalized run-time region will match the mean and standard deviation from training.
	? Tails	Tails matching: Derive the pixel map from the left and right tails calculated during training. The tails are outlying points (representing grey-scale values) on the left and right ends of a histogram beyond which grey-scale values are considered unreliable;

	grey-scale values outside the tails are discarded before the mapping is done. The range of grey-scale values between the tails of the test image’s histogram is linearly mapped to the range between the tails of the training histogram.

PatInspect results

PatInspects reports alignment results identical to PatMax.

In addition the system reports the count and the area of the larger defect found.

6.6 Blob tool

This chapter describes Blob, a vision tool that provides advanced blob analysis of images. This chapter contains the following sections.

The blob tool is used to detect and measure two-dimensional shapes (blobs) within images.

Blob analysis can provide your application with information about the number, location, shape, and orientation of blobs within an image.

Image Segmentation

Since blob analysis is fundamentally a process of analyzing the shape of a closed object, before blob analysis can be performed on an image, the image must be segmented into those pixels that make up the blob being analyzed, and those pixels that are part of the background.

In general, all images that are used for blob analysis start out as grey-scale images of scenes. While it might be easy for a human observer to identify blobs or objects within the scene, before blob analysis can analyze the image, each pixel in the image must be assigned as an object pixel or a background pixel. Typically object pixels are assigned a value of 1 while background pixels are assigned a value of 0.

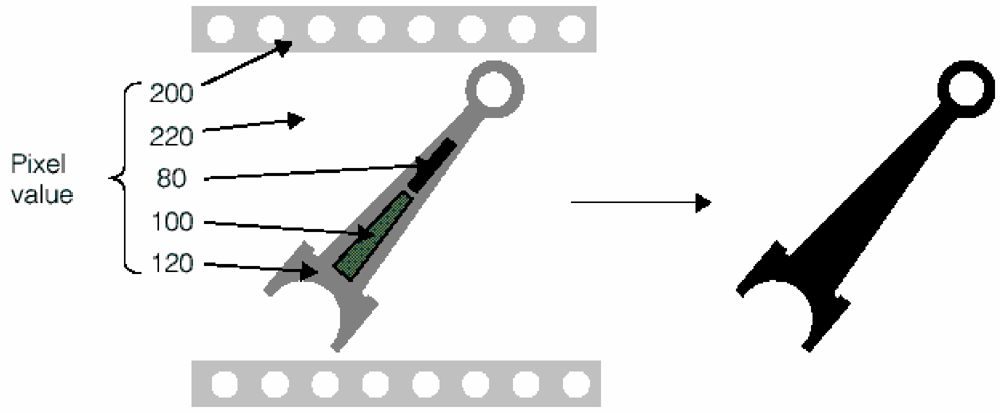
Several techniques can be used to segment images into object pixels and background pixels. The following sections describe these techniques.

Binary Thresholding

The simplest technique for segmenting an image is to pick a threshold pixel value. All pixels with grey-scale values below the threshold are assigned as object pixels, while all pixels with values above the threshold are assigned as background pixels. This technique is called binary thresholding or hard thresholding.

Figure 6-19 shows an idealized scene, the application of a hard threshold, and the resulting segmentation of the image into object and background pixels. In the example below, a threshold value of 150 was used. All pixels with values greater than or equal to 150 are treated as background; all pixels with values less than 150 are treated as object.

Fig. 6-18 Segmenting an image with a binary threshold



Regional thresholding

Binary thresholding, although a simple and fast method has several drawbacks.

Binary threshold works well with high contrast blobs and images with a homogeneous background level.

In addition it cannot cope with variations of light or reflectivity of the parts.

An alternative method of thresholding with better performances in case of non homogeneous background and light variations is called regional thresholding.

With the regional thresholding the level of a pixel is checked against a neighborhood of pixels instead of a fixed threshold.

There are two variants of regional thresholding.

Unsigned difference

With unsigned difference the absolute value of the difference between the pixel level and a weighted average of its neighbors is taken into account. This method provides slightly higher speed but cannot distinguish the color of the blobs. The threshold value has the dimension of a grey level (from 0 to 255).

Signed difference

Signed difference take into account the sign of the difference. Thus it can distinguish blobs having levels above or below the background. The thresholds in this case are two signed values (up to +127 for white blobs, down to –127 for black blobs).

Model difference

Model difference is different from other segmentation techniques in that it uses a model image of the region of interest stored at train time. The difference between the model and the run time image is compared to a fixed threshold. Model difference is the segmentation method more suited to images having a background pattern. However a fine alignment is required to inspect scenes with strong patterns. Consider also the PatInspect tool for this purpose.

Connectivity Analysis

Once an image has been segmented into object pixels and background pixels, connectivity analysis must be performed to assemble object pixels into connected groups of object pixels or blobs.

Connectivity analysis uses connectivity criteria to assemble the object pixels within the image into discrete, connected blobs. In general, connectivity analysis is performed by joining all contiguous object pixels together to form blobs. Object pixels that are not contiguous are not considered to be part of the same blob.

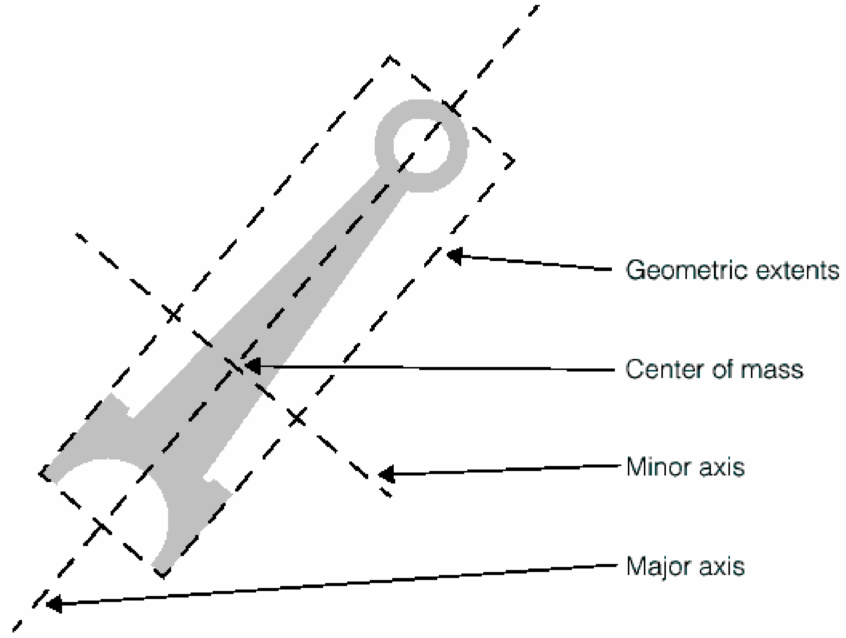
Properties of Blobs

Once an image has been segmented, and the blob or blobs have been located and identified, an application can begin to consider information about the blob or blobs.

A blob is an arbitrary two-dimensional shape. The shape of a blob can be described using a number of different measures. The measures that the Blob tool returns are depicted in the following sections.

Figure 6.20 illustrates the geometric properties of a blob.

Fig. 6-19 Blob geometric properties



Area

The area of a blob is defined as the sum of the weighted pixel values for every nonzero pixel in the blob. If a hard binary threshold was applied to the image to segment it, then the area is simply the number of nonzero pixels in the blob. If grey-scale analysis is being performed, then the area is the sum of the pixel weights.

Perimeter

The perimeter of a blob is the length of the outside edge of the blob. A blob's perimeter is computed by counting the number of pixel edges between pixels with nonzero values and pixels with values of 0.
Because computing the perimeter in this way tends to produce a perimeter that is lager than the actual blob perimeter, a correction factor is applied to the perimeter.

Center of Mass

The center of mass of a blob represents the blob's balance point. If a sheet of a uniform material were cut out in the shape of the blob, the point upon which the blob would balance is the center of mass.

The center of mass is computed by determining the average weighted pixel location for each pixel in the blob in the x-axis and y-axis.

Principal Axes

The major axis of a blob is the axis about which it would be the easiest
to spin the blob. The major axis of a blob always passes through the blob's center of mass. The minor axis of a blob is defined as the axis through the center of mass about which the second moment of inertia is the largest. The minor axis of a blob is the axis about which it would be the hardest to spin the blob. The minor axis of a blob is always a line that is at 90 ??to the major axis passing through the center of mass. Together, the major and minor axes are referred to as the principal axes.

Geometric Extents

The geometric extents of a blob are the distances between the blob's center of mass and the four sides of a rectangle that is oriented to the blob's principal axes and that completely encloses all the blob's pixels.

Inspect or measure mode

The blob tool can be used for two separate purposes.

Inspect mode

Inspect mode verifies the absence of defects. Any segmented pixel is considered a defect. The blob tool allows to specify certain tolerance on the size and number of defects.

Defects are distinguished in "major" and "minor" defects based on their size. Major defects have a size above a major limit. Minor defects have a size below the major limit and above a minor limit. Defects below the minor limit are neglected.

The pass/fail decision is than taken as described by the following table:

Major defect count > 0	Fail
No major defect, Minor defect count > limit	Fail
No major defect, minor defect count < limit	Pass

Measure mode

Measure mode verifies the properties of the blob against a set of tolerances. Note that the properties of a single blob are taken into account. In case of multiple blobs found in the region of interest the larger blob is selected.

Blob settings

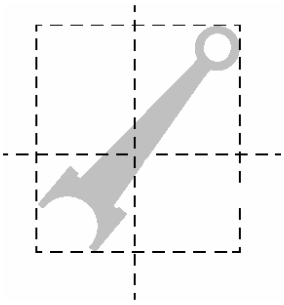
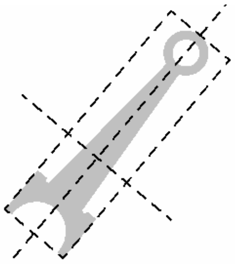
Blobs property page

The Blob page contains the parameters and specification relevant for the reject decision.

Inspect mode

Defect area limit – Tolerable area of all defects	A limit for the total defect area.
Tolerable number of minor defects	Limit for the count of minor defects.
Minor defect area limit	Limit for the size of defect. Defect larger than this limit are deemed as " major defect"

Measure mode

Position offset	Limit for the position offset of the blob fount relative to the train time situation.
Rotation	Limit for the rotation of the blob relative to the train time situation.
Perimeter	Limit for the perimeter size relative to a specified nominal value.
Area	Limit for the blob area relative to a specified nominal value.
Area/Perimeter ratio	Limit for the area/perimeter ratio relative to a specified nominal value. The ratio provides a basic indication of the blob shape.
Axis calculation mode	? Inertia axis – Principal axes of inertia ? Bounding box (un-rotated) – The axes of the bounding rectangle parallel to the x-y axis  ? Bounding box (rotated) – The axes of the bounding rectangle parallel to the principal inertia axes. 
Axis tolerances (H, V)	Tolerances for horizontal (H) and vertical (V) axis.

Threshold page

The threshold page contains settings that influence the segmentation process. The threshold page includes an image panel showing the result of the segmentation. The graphics is updated interactively at any parameter change.

The image panel shows a graphics that change with the setting of an option mode with the following possible choices:

- ? Defects – bounding boxes
- ? Blobs – segmented pixels in red
- ? Threshold – mapped image – red for lighter pixels – blue for darker pixels

Mode (of segmentation)

Mode	Regional thresholding with unsigned difference	Regional thresholding with unsigned difference
	? Unsigned difference	Regional thresholding with unsigned difference
	? Signed difference	Regional thresholding with unsigned difference
Filter size	For regional thresholding mode specifies the size of the neighborhood of pixels taken into account to compute the difference	

Sensitivity

Sensitivity mode	Black	Only "black" pixels, darker than the background are segmented
	White	Only "white" pixels, lighter than the background are segmented
	Black and white	Black and white pixels are segmented with separate thresholds
Black	Threshold for black pixels	
White	Threshold for white pixels	
Min area	Blob size limit. Blobs smaller than this limit are neglected	

Blob results

Inspect mode

? Number of blobs

? Max blob size

Measure mode

The following measurement are reported for the largest blob

? Perimeter

? Area

? Shape factor (Area/perimeter ratio)

? H and V axis size

? Position (x, y of the center of mass)

? Offset (x, y relative to the model position)

? Angle (of major inertia axes)

7. Other tools

7.1 Fitting tools

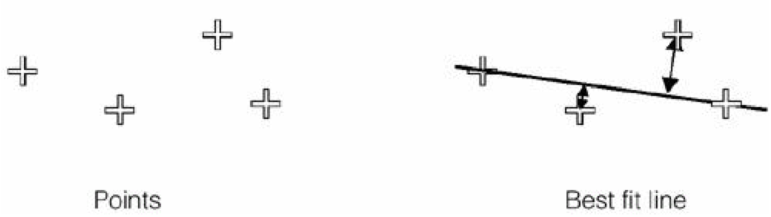
The Fitting tools find the line or circle that best fits a set of points that you supply.

You use the Fitting tool to locate lines or circles based on individual point locations that you determine using other vision tools.

How The Fitting Tool Works

Fitting a Line

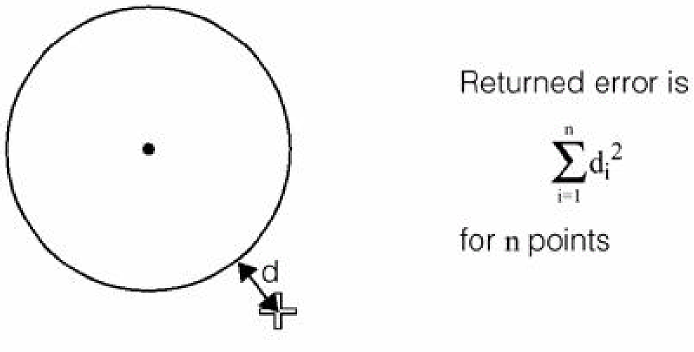
The Line tool considers the best fit line by computing the sum of the squares of the minimum distance from each point to the line. The best fit line is the single line that has the lowest value for this computation.



The result of the Line tool is the center position and the orientation of the line.

Fitting a Circle

The Circle tool lets you fit a set of points to a circle by minimizing the absolute value of the signed square distance error. This method computes the radius of the fitted circle.



The result of the circle tool is the center and the size (diameter) of the circle.

RMS Error

The Fitting tools provide with the RMS (root mean square) error an indication about the quality of the fitting.

The RMS error is the square sum of the distances between the actual positions of the points and their predicted values, divided by the number of points.

7.2 Distance tools

Distance tools calculates the distance between points or between a point and a line.

The Point Distance tool takes two locating tools as input and returns the distance between them.

The Line distance tool takes a line and a point locating tool as input and returns the orthogonal distance.

8. Software Installation

The ProInspect software is likely to be already installed in your system. This section provides instructions in case you should install the ProInspect software from scratch using the installation CD provided with your system.

The PC must be prepared with the following hardware and software before installing ProInspect:

? Microsoft Windows NT 4.0 + Service Pack 5 or higher
? Cognex 8100 PCI image acquisition module
? Microsoft Windows Installer

If your system does not already have Widows Installer the Instwmsi.exe install program is provided in the principal CD folder.

ProInspect components are provided as “Microsoft Windows Installer” (.msi) packages. The installer is started automatically by double clicking the install file. Windows Installer will then guide you through the installation process.

A typical ProInspect installation requires the following components:

? Cognex 8100 driver
? Parallel Port Driver
? ProInspect core program

Cognex 8100 driver installation

A suited driver for your 8100 hardware must be installed. The installation package is \Driver\WINNT4\Cognex 8100M.msi

Parallel Port Driver (Tinyport)

Your system might require a special parallel port driver if the parallel port is used to interface the system. The parallel port driver installation package is \Driver\WINNT4\Tinyport.msi

ProInspect Installation

The core ProInspect installation is the ProInspect 2.4.msi file in the principal CD folder.