Sherlock 7 Technical Resource

Teledyne DALSA Corp., Industrial Products (IPD)
www.teledynedalsa.com/ipd   978.670.2002 (U.S.A.)

Document Revision: August 9, 2012

Search Line Based

Search utilities

A common task in machine vision applications is to train a feature in one image and search for it in other images. This may be done for several reasons:

- To verify that a part (resistor, bolt, label, pill) is present
- To determine how well an instance of the feature in an image matches a “golden template” instance of the feature
- To determine the location of a feature relative to other features

Sherlock 7 includes these search tools:

	Strengths	**Limitations**
Correlation Creates a map of the grayscale values within the ROI	Accurate and fast Very good at coping with changes in illumination and focus Can find multiple matches	Allowing for image scaling and rotation can increase execution time significantly
Geometric Creates a model of edges in the feature	Finds matches at any rotation Can find multiple matches	Does not allow for image scaling
Line Based Creates a model of straight edges in the feature	Optimized to look for straight edges Finds matches at any rotation Allows for scaling Generally runs faster than **Geometric**	Can find only one match
Verify Pattern An expansion of Search Correlation that can be trained on more than one pattern.	Can be trained on more than one pattern Works in color images	Can search for only one of its trained patterns at a time

⚠ **Search Line Based** works from Rectangle ROIs only.

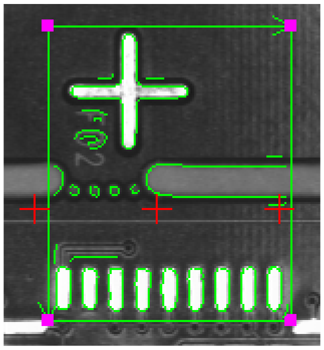
⚠ **Search Line Based** works on monochrome images only.

Search Line Based creates a model of the edges in the feature, then looks for the same edges in new images.

To use **Search Line Based**

1. Train a good example of the feature
2. Define the search area
3. Acquire a new image, search for the feature and get its location, match score, and other information

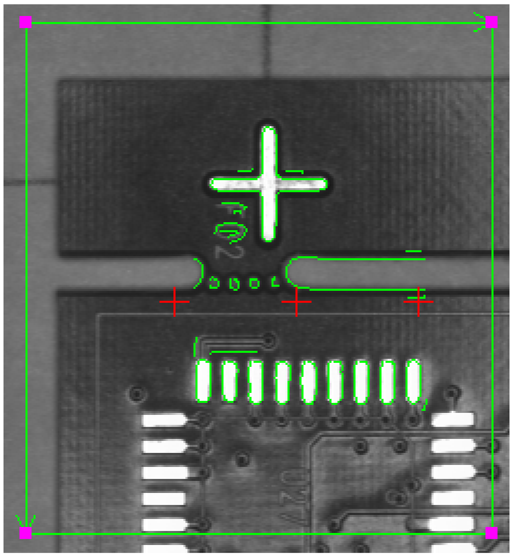
Train the feature



Surround the feature with a rectangular ROI and select the **Search Line Based** algorithm from the ROI's algorithm list. As soon as the algorithm is selected, the edges are trained.

After the edges are trained, the algorithm is automatically put into run (search) mode.

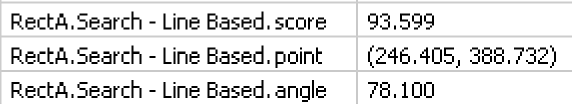
Define the search area



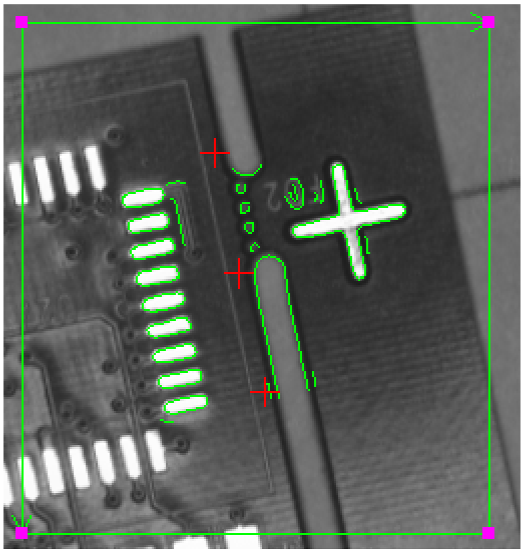
Expand the same ROI to define the search area. A smaller search area results in less processing time, so restrict the search area to that part of the image where you know the pattern will occur.

Acquire a new image and search

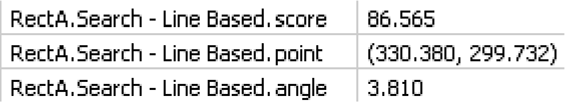
In run mode, **Search Line Based** finds the best match for the set of edges, and returns the location and score of the match. **Search Line Based** searches the full 360 degrees to find the best match to the trained edges.



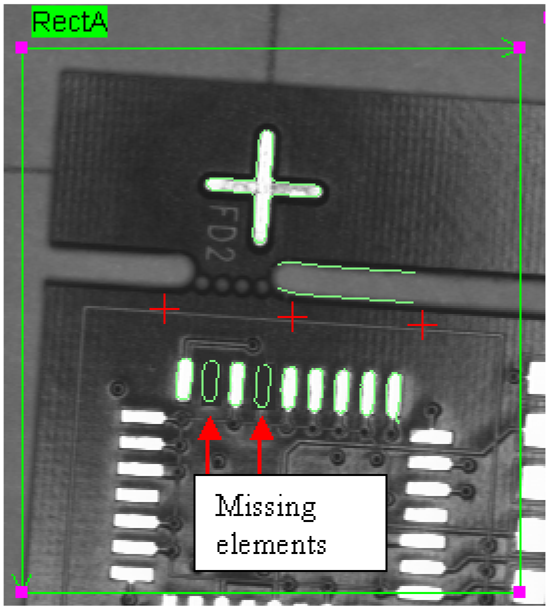
RectA.Search - Line Based.score	93.599
RectA.Search - Line Based.point	(246.405, 388.732)
RectA.Search - Line Based.angle	78.100



Here a less-than-perfect match was found; note the lower score.



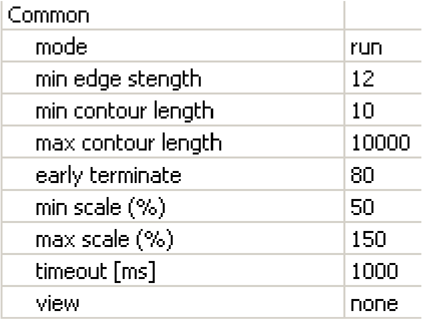
RectA.Search - Line Based.score	86.565
RectA.Search - Line Based.point	(330.380, 299.732)
RectA.Search - Line Based.angle	3.810



Missing elements

Run (search) parameters

The **Common** parameters (which are, in fact, the only parameters) allow you to control how the algorithm finds a match.



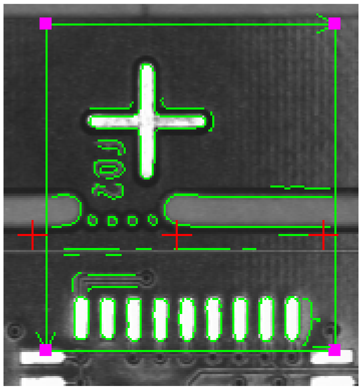
Common	
mode	run
min edge stength	12
min contour length	10
max contour length	10000
early terminate	80
min scale (%)	50
max scale (%)	150
timeout [ms]	1000
view	none

mode – You can train a new feature model by changing **mode** to **train**, then immediately back to **run**. The retrain occurs as soon as you change **mode** to **train**; you do not have to execute the investigation or click the **Parameter** dialog's **Apply** button.

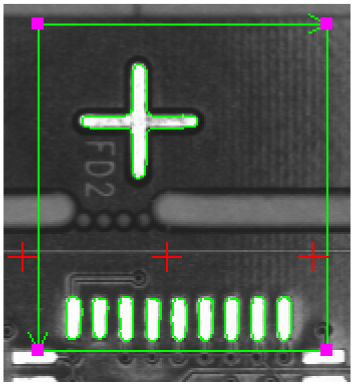
⚠ If you set **mode** to **train** but forget to set it back to **run** before running the investigation, every execution of the algorithm will result in a retraining!

view – If **pattern**, the edges from the trained feature are displayed in green. If **scene**, the edges found in the ROI are displayed in red. If **pattern and scene**, all edges are displayed. If **none**, no edges are displayed. Selecting one of the options is a matter of viewer preference only; it does not affect algorithm execution. If **view** is set to **pattern** or **pattern and scene** and no match is found, the trained feature edges are drawn at the upper-left corner of the ROI.

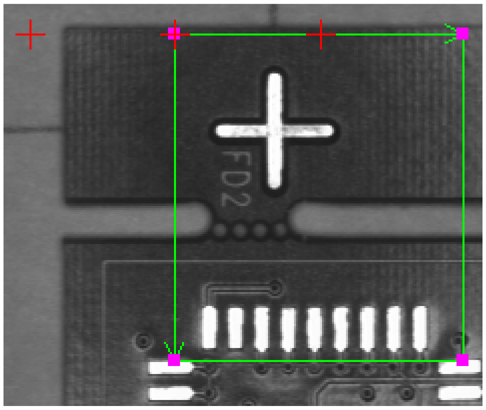
min edge strength – At train time, only edges that are least as strong **min edge strength** are included in the feature model.



min edge strength = 10
mode = **train**



min edge strength = 30
mode = **train**

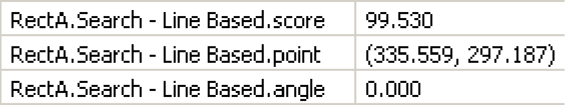


min edge strength = 100
mode = **train**
No edges trained. Note the positions of the pattern-location markers (red crosses).

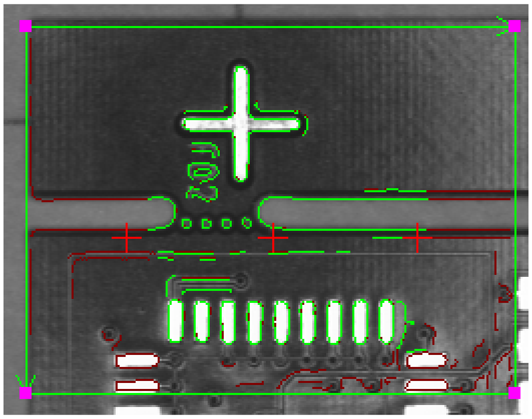
At run time, only edges in the ROI that are least as strong as **min edge strength** are compared to the edges in the feature model.

This parameter can be set to different values at train and run time. In the following examples, the features trained with **min edge strength** = 10 were used. **View** = **pattern and scene**.

runtime **min edge strength** = 10
Good match

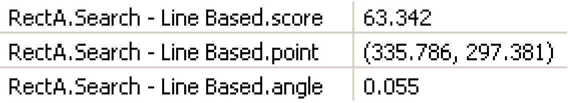


RectA.Search - Line Based.score	99.530
RectA.Search - Line Based.point	(335.559, 297.187)
RectA.Search - Line Based.angle	0.000

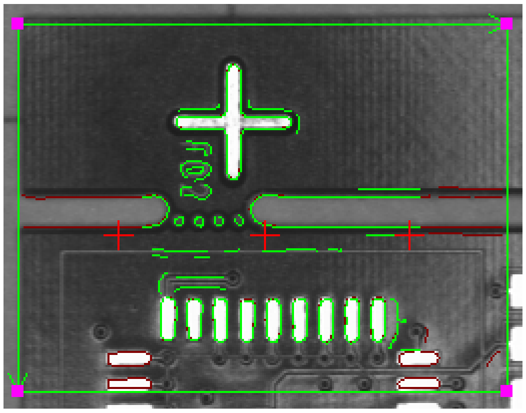


runtime **min edge strength** = 20

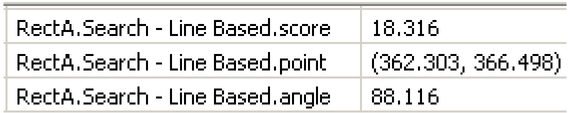
What cannot be seen in this image is that some of the edges that are in the feature model are not found at run time, so the score is lower.



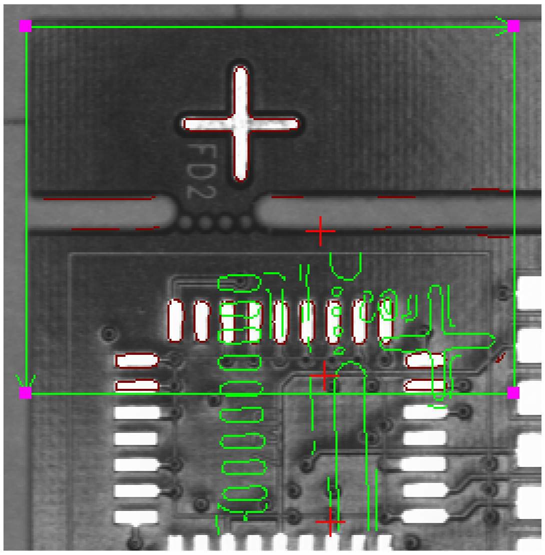
RectA.Search - Line Based.score	63.342
RectA.Search - Line Based.point	(335.786, 297.381)
RectA.Search - Line Based.angle	0.055



runtime **min edge strength** = 25
Not a real match.



RectA.Search - Line Based.score	18.316
RectA.Search - Line Based.point	(362.303, 366.498)
RectA.Search - Line Based.angle	88.116

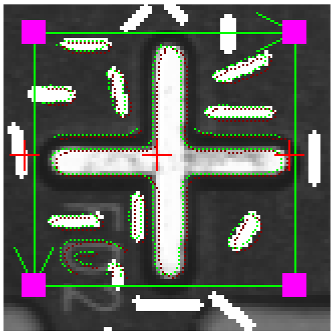


min contour length – At train time, edges that are composed of fewer than **min contour length** points are not included in the feature model. At run time, edges in the ROI that are composed of fewer than **min contour length** points are not compared to the feature model.

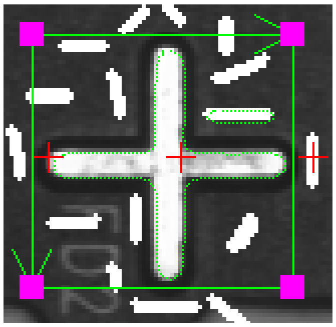
max contour length – At train time, edges that are composed of more than **max contour length** points are not included in the feature model. At run time, edges in the ROI that are composed of more than **max contour length** points are not compared to the feature model

Only edges that were included in the feature model at train time are taken into consideration when determining how well the feature in a new image matches the model. Edges that are in a new image but not in the model do not affect the match score. However, edges that are in the model that are not found in a new image can affect (lower) the match score.

mode = train



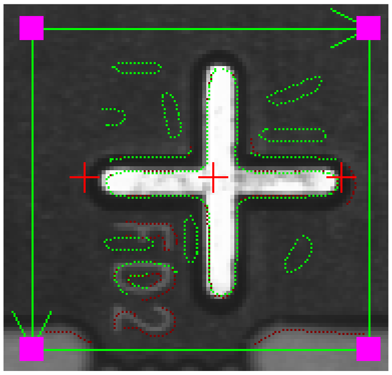
min edge strength = 10
min contour length = 10
Small edges included
in model
Feature model 1

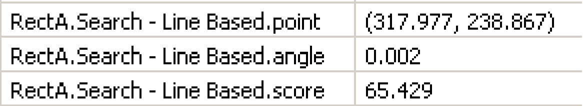


min edge strength = 10
min contour length = 35
Small edges not included
in model
Feature model 2

Trained: Feature model 1
Mode = **run, min edge strength** = 10, **min contour length** = 10
View = pattern and scene

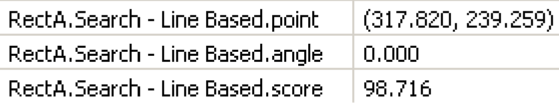
When applied to a feature that is missing some of the edges in the model, the algorithm finds a match, but with a lower score.



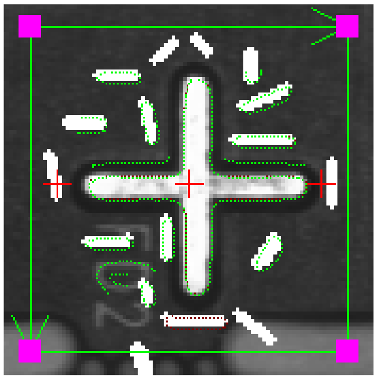


RectA.Search - Line Based.point	(317.977, 238.867)
RectA.Search - Line Based.angle	0.002
RectA.Search - Line Based.score	65.429

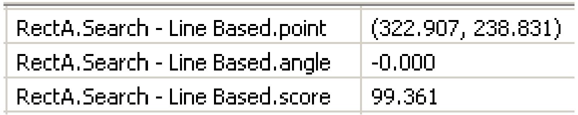
Trained: Feature model 1
Runtime **min edge strength** = 10, **min contour length** = 35
View = pattern and scene



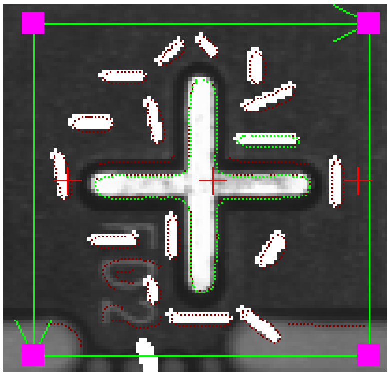
RectA.Search - Line Based.point	(317.820, 239.259)
RectA.Search - Line Based.angle	0.000
RectA.Search - Line Based.score	98.716



Trained: Feature model 2
Runtime **min edge strength** = 15, **min contour length** = 10
View = **pattern and scene**



RectA.Search - Line Based.point	(322.907, 238.831)
RectA.Search - Line Based.angle	-0.000
RectA.Search - Line Based.score	99.361



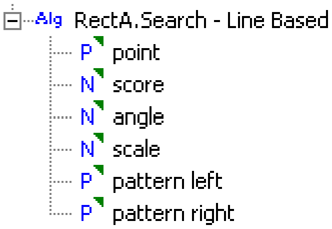
There are more edges in the ROI than in the feature model, but the algorithm just looks for matches between edges in the feature model and edges in the ROI, so the algorithm returns a high score.

early terminate – As soon as a pattern with this score or greater is found, the search terminates. This may mean that another instance of the pattern with a greater score is ignored.

min scale (%) – The minimum the feature can scale down in the image and still match the feature model.

max scale (%) – The maximum the feature can scale up in the image and still match the feature model.

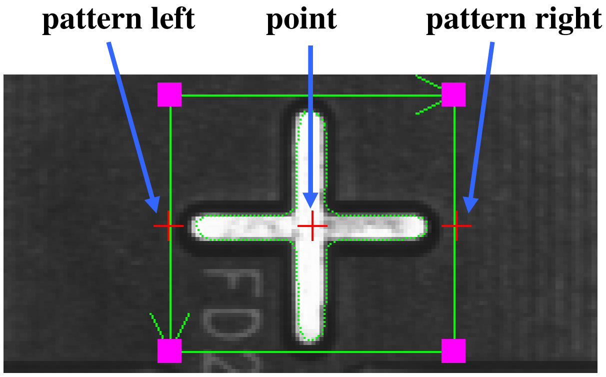
Returned values (readings)

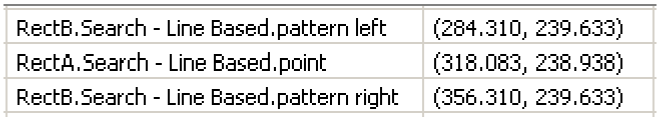


Alg RectA.Search - Line Based
- P point
- N score
- N angle
- N scale
- P pattern left
- P pattern right

point – The location of the best match
score – The score of the best match. A perfect match will return a score of 100.
angle – The angle of the best match. The angle is returned in degrees or radians, depending on the setting in Sherlock's **Options → Application** dialog.
scale – The scale of the best match, as a percentage of the trained feature.

pattern left and **pattern right** – Two points created by the algorithm, centered about **point** and separated by the width of the ROI when the feature is trained. **Pattern left** and **pattern right** can be used as landmarks in an alignment scheme that needs to calculate rotation.





RectB.Search - Line Based.pattern left	(284.310, 239.633)
RectA.Search - Line Based.point	(318.083, 238.938)
RectB.Search - Line Based.pattern right	(356.310, 239.633)