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 Correlation

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:

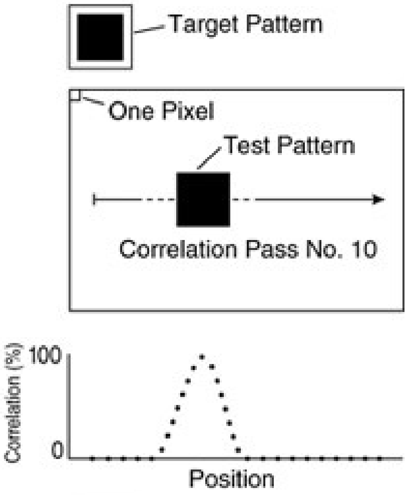
	Strengths	**Limitations**
Correlation Creates a map of the grayscale values within the ROI	Accurate and fast Very good at coping with changes in illumination and focus Can find multiple matches	Allowing for image scaling and rotation can increase execution time significantly
Geometric Creates a model 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
Edge Creates a model of edges in the feature **Note: Search Edge** requires an additional license. It is not part of the standard Sherlock release.	Very accurate Allows for image scaling and rotation Can be trained on more than one pattern	
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

⚠ The search tools work from Rectangle ROIs only.

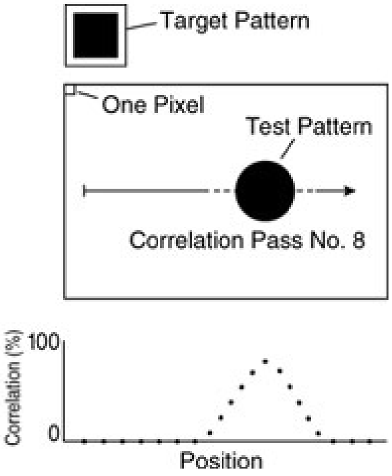
⚠ The search tools work on monochrome images only, with the exception of **Verify Pattern**.

Search Correlation creates a map of the grayscale values of the feature, then looks for matches for the map in the region of interest (ROI).

Suppose you have trained a black square as the feature. **Search Correlation** mathematically "scans" the map (target pattern) across the ROI one pixel at a time. When the map is scanned across the white area, the algorithm produces a low score. The score increases as the map starts to cross the black square in the ROI, and the score reaches a maximum when the squares overlap precisely. The maximum score indicates the position at which the patterns exhibit the best match.



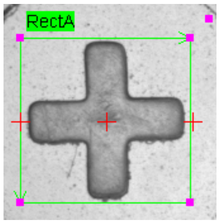
Suppose the ROI contains a black circle rather than a black square. The trained feature would correlate fairly well, although not perfectly, with the black circle.



To use **Search Correlation**

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

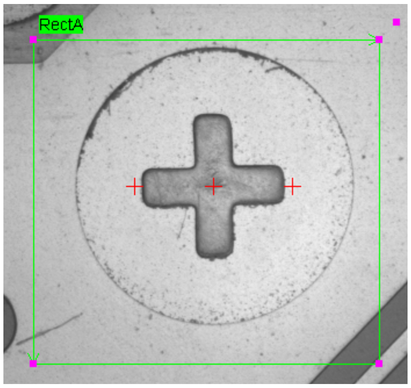
Train the feature



Surround the feature with a rectangular ROI and select the **Search Correlation** algorithm from the ROI's algorithm list. As soon as you select the algorithm, the feature is automatically trained. The algorithm constructs a map of the gray-scale pixel values within the ROI.

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

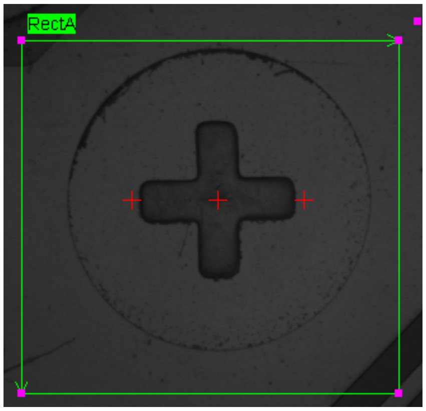
Define the search area



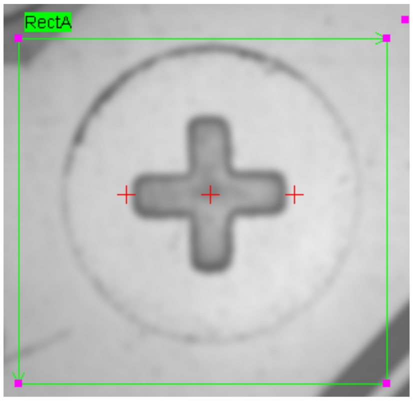
Expand the same ROI to define the search area. A smaller search area results in less processing time, so restrict the search area to that part of an image where you know the pattern will occur.

Acquire a new image and search

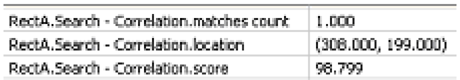
In run mode, **Search Correlation** compares the grayscale values in the map and the grayscale values in the ROI, in such a way that changes in illumination and image contrast have little effect on the score of the match. The algorithm is also insensitive to changes in focus and other forms of image degradation, up to a point.



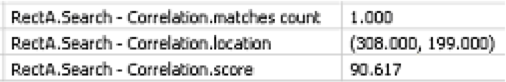
Decreased illumination



Defocused image

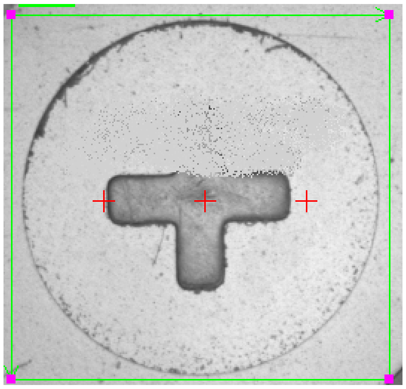


RectA.Search - Correlation.matches count	1.000
RectA.Search - Correlation.location	(308.000, 199.000)
RectA.Search - Correlation.score	98.799

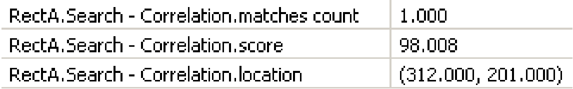


RectA.Search - Correlation.matches count	1.000
RectA.Search - Correlation.location	(308.000, 199.000)
RectA.Search - Correlation.score	90.617

Of course, in a robust setup, the lighting and focus should never vary this much!



Incomplete feature



RectA.Search - Correlation.matches count	1.000
RectA.Search - Correlation.score	98.008
RectA.Search - Correlation.location	(312.000, 201.000)

Train parameters

Search Correlation cannot, by default, find a pattern that has rotated more than a few degrees from the rotation at which it was trained (unless the pattern exhibits some form of symmetry), or changed in scale more than a small percentage from the scale at which it was trained.

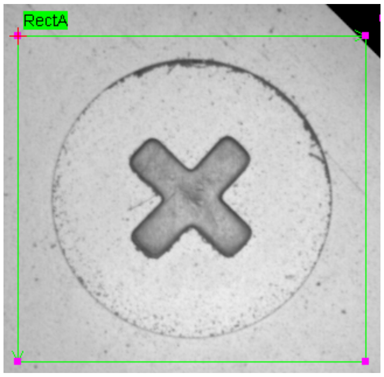


Image rotated 45 degrees

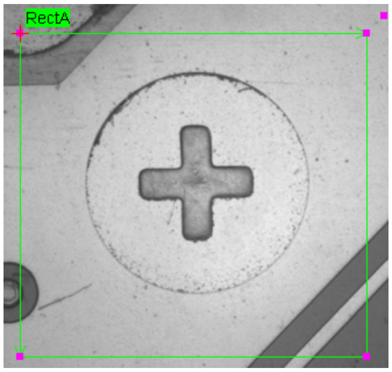
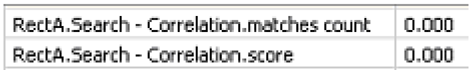
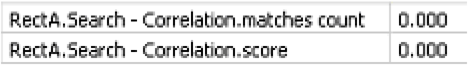


Image scaled to 80% of original



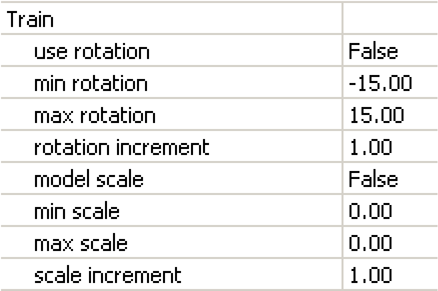
RectA.Search - Correlation.matches count	0.000
RectA.Search - Correlation.score	0.000



RectA.Search - Correlation.matches count	0.000
RectA.Search - Correlation.score	0.000

If you know that the pattern can rotate and/or change scale from image to image, you will have to enable rotation and/or scaling, and set their parameters <u>before you train the pattern</u> (by changing **execution mode** to **Train**; see below).

For every combination of the selected rotation and scaling factors, **Search Correlation** creates an instance of the pixel map.



Train	
use rotation	False
min rotation	-15.00
max rotation	15.00
rotation increment	1.00
model scale	False
min scale	0.00
max scale	0.00
scale increment	1.00

use rotation – If **True**, the next three parameters are used to create instances of the pixel map at different rotations.

min rotation – The greatest counterclockwise angle of rotation at which the search will find the feature. Expressed as a negative angle from 0 to -180.

max rotation – The greatest counterclockwise angle of rotation at which the search will find the feature. Expressed as a positive angle from 0 to 180.

rotation increment – The number of degrees between each instance of the pixel map.

model scale – If **True**, the next three parameters are used to create instances of the pixel map at different scale factors.

min scale – The smallest scaling factor at which the search will find the feature. Expressed as a multiplier, where 1.0 is no downward scaling.

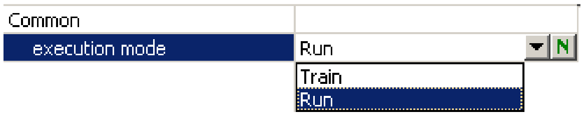
max scale – The largest scaling factor at which the search will find the feature. Expressed as a multiplier, where 1.0 is no upward scaling.

scale increment – The change in scale between each scaled instance of the pixel map.

For example, if you set **use rotation** to **True**; change **min rotation** to –45; **max rotation** to 45; and **rotation increment** to 5; then retrain the pattern, **Search Correlation** will create 18 versions of the pixel map: –45 to 45 is a range of 90 degrees, divided by a rotation increment of 5. At run time, the algorithm will find the best match among all 18 versions.

If you also set **model scale** to **True**; **min scale** to 0.50; **max scale** to 1.40; and **scale increment** to 0.10; then retrain the pattern, **Search Correlation** will create 180 versions of the pixel map (18 rotations x 10 scales). This can increase execution time significantly!

Note: To retrain the **Search Correlation** pattern, which you must do anytime you change any of the **Train** parameters, change **execution mode** to **Train**, then immediately back to **Run**. The retrain occurs as soon as you change **execution mode** to **Train**; you do not have to execute the investigation or click the **Parameter** dialog's **Apply** button.



Common

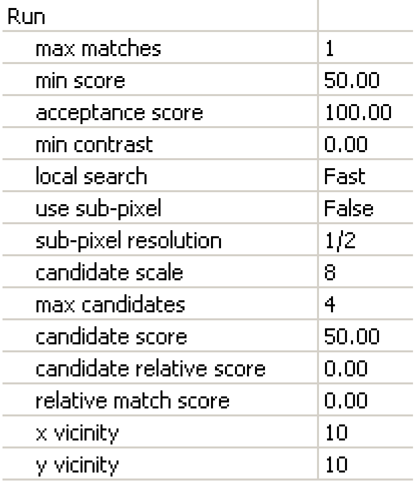
execution mode Run

⚠ If you set **execution mode** to **Train**, be sure to set it back to **Run** before exiting the **Parameters** dialog. If you leave the algorithm in **Train** mode and run the investigation, every execution of the algorithm will result in a retraining!

Run (search) parameters

Search Correlation has several parameters that you can set to control how the algorithm searches for a feature.

Search Correlation usually works quite well with the other parameters left at their defaults. Except for **max matches** and **min score**, you will rarely need to change any of the **Run** parameters. For some of the parameters, setting them to values other than their defaults requires detailed knowledge of the contents of the trained feature and the acquired images.



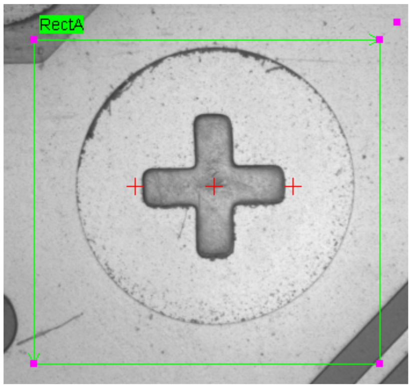
Run	
max matches	1
min score	50.00
acceptance score	100.00
min contrast	0.00
local search	Fast
use sub-pixel	False
sub-pixel resolution	1/2
candidate scale	8
max candidates	4
candidate score	50.00
candidate relative score	0.00
relative match score	0.00
x vicinity	10
y vicinity	10

max matches – By default, the algorithm finds one (the best) match for the feature. Change this parameter if you want to find more than one match.

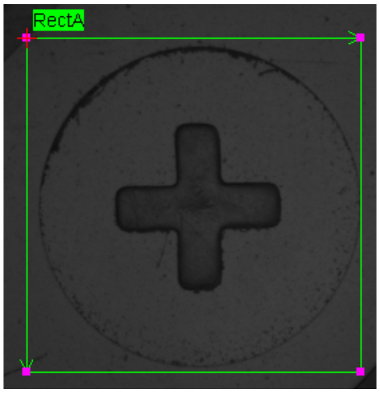
min score – Only matches with scores equal to or greater than **min score** are returned.

acceptance score – The score above which a pattern is always returned as a match. The search is terminated as soon as **max matches** patterns with scores above this value are found. For example, if **max matches** is set to 5 and **acceptance score** is set to 85, as soon as 5 matches with scores of 85 or greater are found, the search is terminated, even though there may be other instances of the pattern with higher scores in the ROI. This can result in decreased processing time, but it can also result in the skipping of patterns with scores greater than those returned as matches. If **acceptance score** is set to 100 (the default), the entire ROI is searched and the very best matches are returned.

min contrast – The minimum contrast required to produce a match. It is the minimum standard deviation of the pixel values in a possible match relative to the standard deviation of the pixel values in the trained pattern. This parameter is usually set to a value between 0 and 1, although values greater than 1 can be used if the search area has greater contrast than the trained pattern. Setting this parameter to a value other than its default of 0 prevents the algorithm from matching the trained feature to an instance of the feature whose brightness or contrast differs significantly.



Standard deviation of trained feature = 41.0



Standard deviation of best potential match in decreased illumination image = 10.7

10.7 / 41.0 = 0.26
if **min contrast** = 0 (default), 0.1 or 0.2 : match found
if **min contrast** = 0.3 : match not found (as shown)

use sub-pixel – If set to **True**, coefficients required for sub-pixel pattern matching will be calculated, allowing the match locations to be refined to sub-pixel precision.

sub-pixel resolution – The resolution of the sub-pixel refinement of the match locations: ½, ¼, 1/8, 1/16, 1/32, or 1/64. This parameter is only used when **use sub-pixel** is set to **True**.

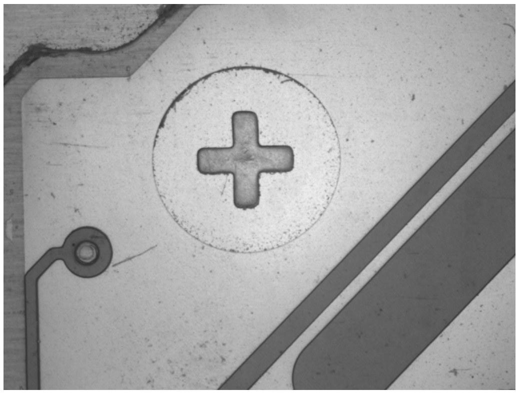
relative match score – The relative rejection threshold for final matches. The relative rejection threshold is determined by multiplying **relative match score** by the match score of the best match found. If set to a value other than its default of 0 (zero), **relative match score** is used instead of **min score** to determine whether a match is returned as a final match.

x vicinity, y vicinity – Once a match has been found, **Search Correlation** will not look for other instances of the pattern **x** pixels left and right and **y** pixels above and below the center of the match.

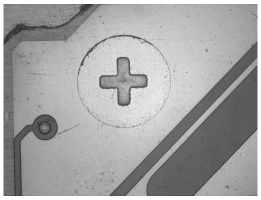
In run mode, **Search Correlation** is a two-pass algorithm. In the first pass, the algorithm performs a low-resolution search to identify candidates for a match. In the final pass, the algorithm revisits the candidates and examines them in more detail to determine which are valid matches. There are several parameters that you can set to control how candidates are found.

local search – The method used to search the local neighborhoods of candidates found during the low-resolution search. When the parameter is set to **Fast** a binary hill-climbing method is used. The hill-climbing method can get stuck in a local maximum, in particular when the target and model images contain fine regular patterns, such as closely spaced vertical lines. To prevent this from occurring, the parameter can be set to **All**, in which case an exhaustive search of the neighborhood is performed. The exhaustive search is slower for large **candidate scale** factors.

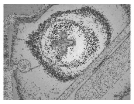
candidate scale – The resolution reduction factor for the initial low-resolution search: 2, 4, 8, 16, or 32. This determines how much the image and trained pattern will be scaled down to locate candidate matches on the first pass of the search. You can think of this as "zooming out" the image and trained pattern.



Original image

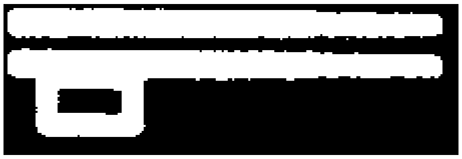


candidate scale = 2

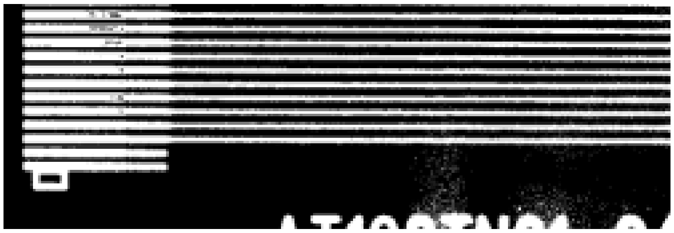


candidate scale = 4

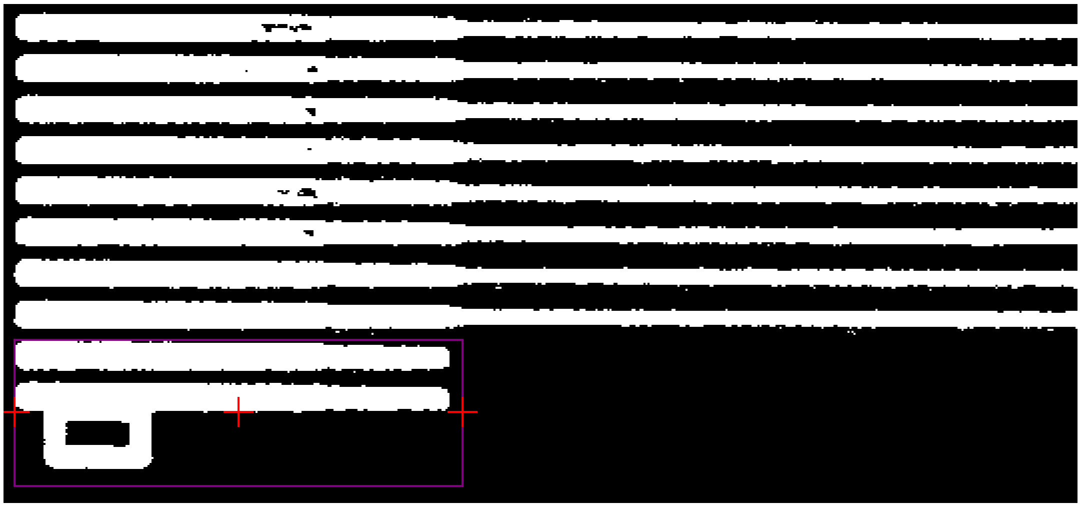
A higher **candidate scale** may decrease search time, but may also result in false matches or no matches if it causes distinguishing features to be removed during the first, low-resolution pass.



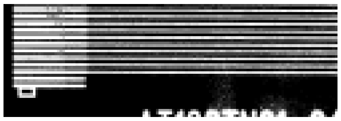
Trained pattern



candidate scale = 4;



Good final match



candidate scale = 8 ;

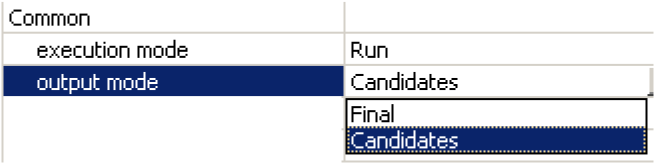
No final match

max candidates – The maximum number of candidates that will be passed to the final pass. **Max candidates** should be greater than **max matches**, since the final matches are selected from among the candidates.

candidate score – Only candidates with scores equal to or greater than **candidate score** are passed to the final pass.

candidate relative score – The relative rejection threshold for candidate matches. The relative rejection threshold is determined by multiplying **candidate relative score** by the match score of the best candidate match found. If set to a value other than its default of 0 (zero), **candidate relative score** is used instead of **candidate score** to determine which candidates are passed to the final pass of the algorithm.

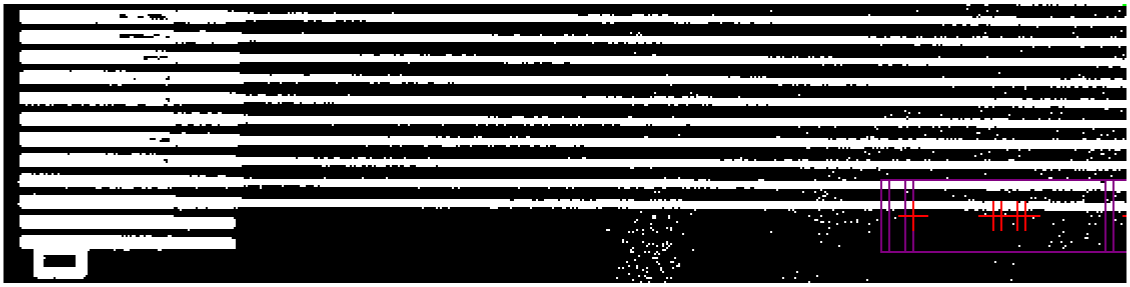
If you change **output mode** to **Candidates, Search Correlation** performs only its first pass and displays the candidate matches. This may be useful if the search area contains two or more patterns that are potential matches for the map, and you need to determine how to set the search parameters to return the match you want; or if the final pass returns no matches, even though you know there is a good match within the ROI.



Common

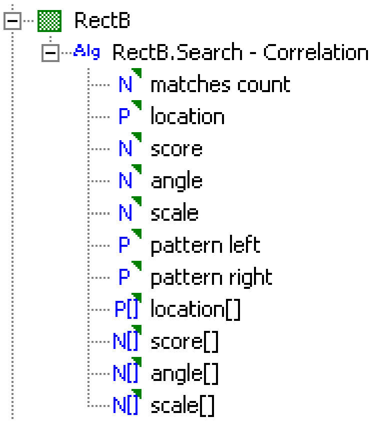
execution mode Run

output mode Candidates



candidate scale = 8, **output mode** = **Candidates**
(image not at original scale)

Returned values (readings)

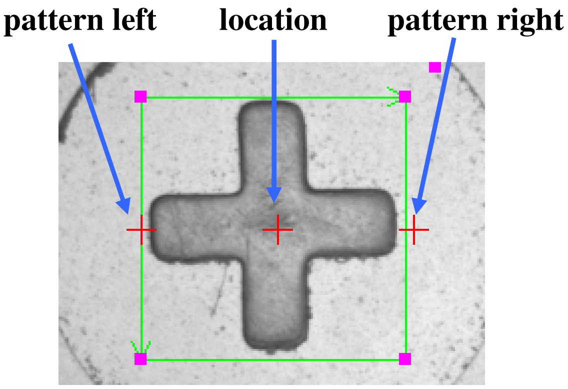


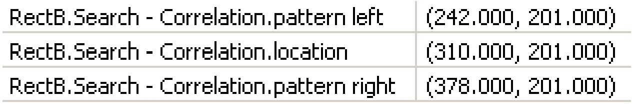
RectB
Alg RectB.Search - Correlation
N matches count
P location
N score
N angle
N scale
P pattern left
P pattern right
P[] location[]
N[] score[]
N[] angle[]
N[] scale[]

matches count – The number of matches, up to a maximum of **max matches.**
location – The location of the best match.
score – The score of the best match. A perfect match returns 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 **location** 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.

If **max matches** is greater than 1 and more than one match is found:
location[ ] – The locations of the matches, ordered according to **score[ ]**.
score [ ] – The scores of the matches, ranked from highest to lowest.
angle[ ] – The angles of the matches, ordered according to **score[ ]**.
scale[ ] – The scales of the matches, ordered according to **score[ ]**.

The first element of each array contains the same information as its corresponding scalar reading. (E.g., **location[0]** = **location**.)





RectB.Search - Correlation.pattern left	(242.000, 201.000)
RectB.Search - Correlation.location	(310.000, 201.000)
RectB.Search - Correlation.pattern right	(378.000, 201.000)