NHL STENDEN
computer vision & data science

Computer Vision

Blob matching

10 April 2018

j.van.de.loosdrecht@nhl.nl, jaap@vdlmv.nl

Blob matching

Overview:

- **Introduction**
- **Demonstrations**
  - **Creating a Blob Matcher (BM)**
  - **Matching a blob**
  - **Evaluating a BM with a Class Image Set (CIS)**
  - **Using a BM with a blob group (*)**
- **Exercises**
- **Demonstrations**
  - **FindPatterns: Finding all patterns in an image**
  - **FindBlob: Finding one blob in an image**
- **Comparing with alternative methods**
- **Using in scripts (*)**
- **Class Lexicon (*)**

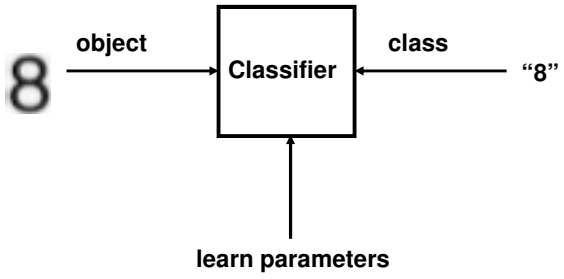
Introduction classification

Classification:
the assignment of an object characterized by a set of features to one of a number of predefined classes.

Example: Optical Character Recognition

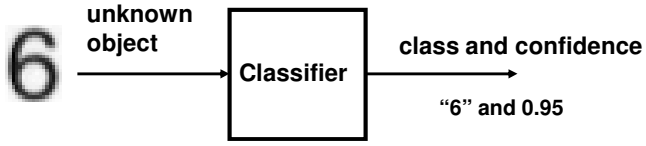
Introduction classification

- **Learning mode**
  - **Learn set: set of objects with known classes**



Introduction classification

- **Production mode**



27-aug-18 Blob matching 5

Introduction blob matching

There are several methods for implementing a classifier:

- **Blob matching**
- **Neural networks**
- **Correlation**
- **etc**

27-aug-18 Blob matching 6

Introduction blob matching

Idea:

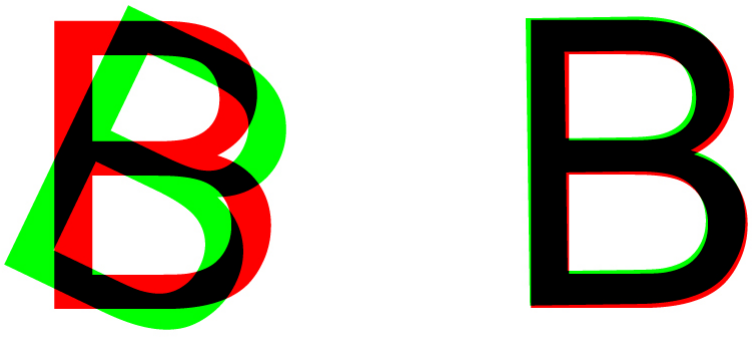
- **Training:**
  - **add one example pattern for each class**
- **Matching:**
  For all patterns do
  - **Resize blob to same size as pattern**
  - **For n rotations do**
    - **Rotate resized blob**
    - **Calculate difference by mapping center of gravities**

  Best match is pattern with the smallest difference

27-aug-18 Blob matching 7

Introduction blob matching

Calculate differences after normalizing for size and alignment on center of gravity

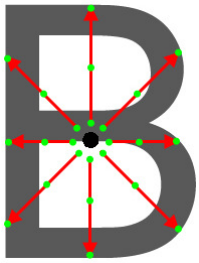


27-aug-18 Blob matching 8

Introduction blob matching

Calculating the differences:

- **Contour matching: using the contour pixels only (fast)**
- **Filling matching: the interior of the object is sampled to find more accurate match (slower)**
  example: nr of rotations = 8, fillSampleSize = 3



27-aug-18 Blob matching 9

Introduction blob matching

Filling matching has 2 parameters:

- **fillSampleSize**
  the number of samples on each rotation ray
- **perimeterFillRatio**
  the ratio of the contribution between the error calculated by the contour match and the filling matching
  total error = contour error + perimeterFillRatio * filling error

Examples:

- **perimeterFillRatio = 0: only contour matching**
- **perimeterFillRatio = 1: both contour and filling matching with equal weights**
- **perimeterFillRatio = 100: "only" filling matching**

27-aug-18 Blob matching 10

Introduction blob matching

The Blob Matcher is a tool which finds the best match for a blob from a list of blobs. For good operation it is required that the blobs can be segmented reliable

The implementation of the Blob Matcher in Vision Lab uses a combination of 3 techniques:

- **Contour matching, mandatory**
  Parameter: number of rotations
  Only the contour pixels are used to find the scaling and rotation
- **Filling matching, optional**
  Parameters: perimeterFillRatio and fillSampleSize
  The interior of the object is sampled to find more accurate scaling and rotation
- **Random check, optional, *normally NOT used, default value = 0***
  Parameter: randomSampleSize
  Maximum number of pixels which are selected to check the matching

27-aug-18 Blob matching 11

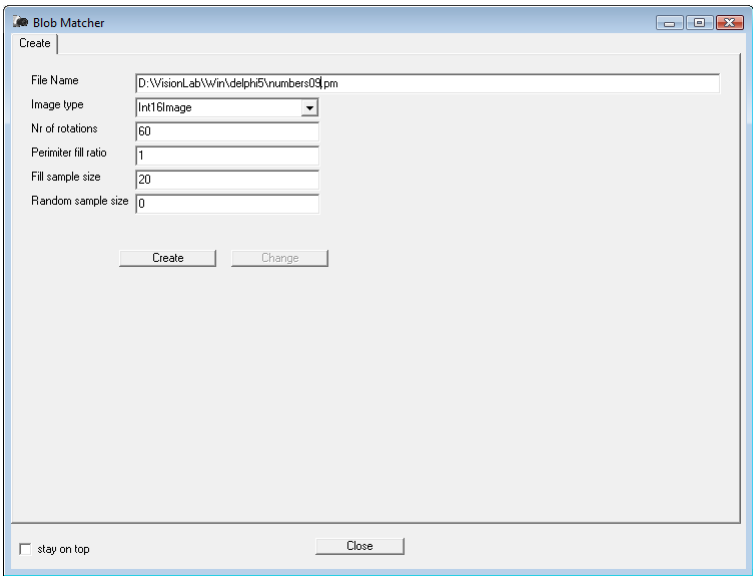
Demonstration creating a blob matcher

- **Select "Create Blob Matcher" from File menu**
  - **Image type: Int16Image**
  - **Nr of rotatiuons: 60**
  - **Perimeter fill ratio: 1**
  - **Fill sample size: 20**
  - **Random sample size: 0**
  - **Click create button**
- **Open image gray09.jl, image with the templates**
- **ThresholdIsoData image**
- **Use ROI to crop '0' in separate image**
- **2nd select image with ´0´**
- **Type in ´zero´ in field right to Add pattern button**
- **Click Add pattern button**
- **Add the other nine numbers**

- **(use BM_09.pm for complete matcher without adding all digits manual)**

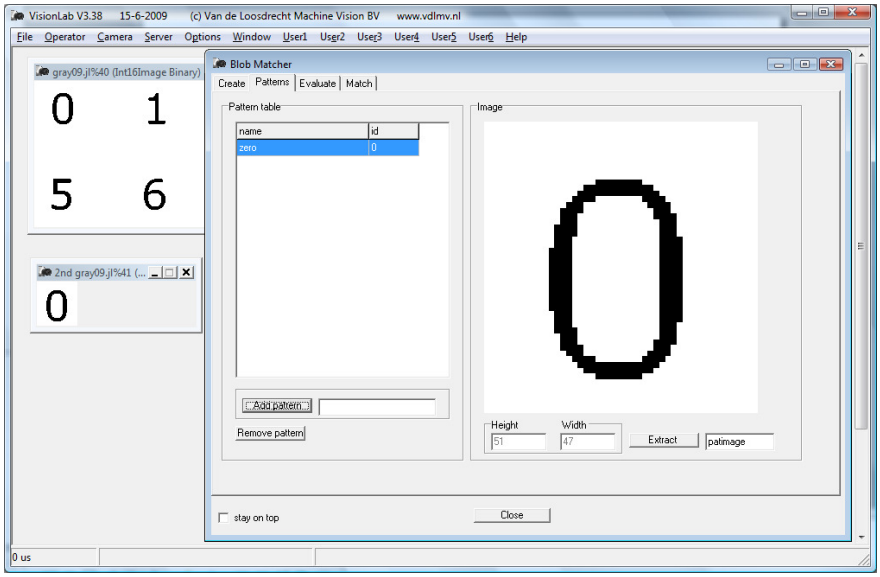
27-aug-18 Blob matching 12

Create blob matcher



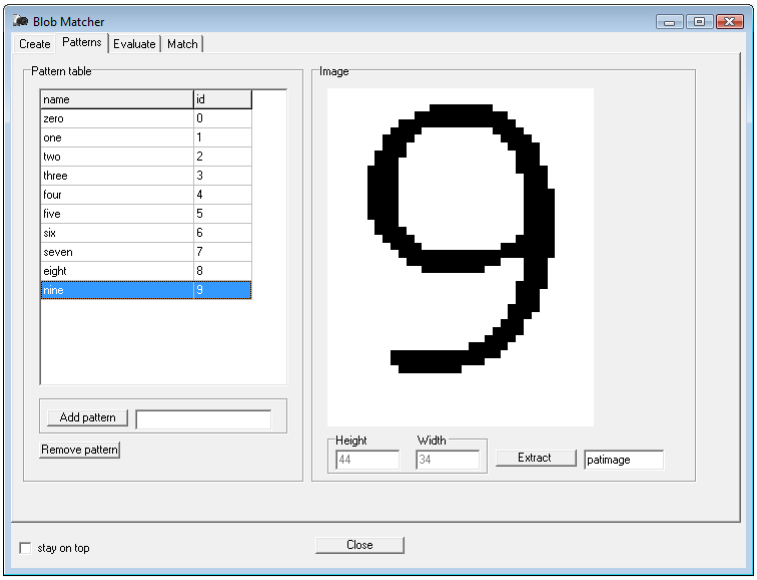
27-aug-18 Blob matching 13

Add first pattern



27-aug-18 Blob matching 14

After adding the 10th pattern



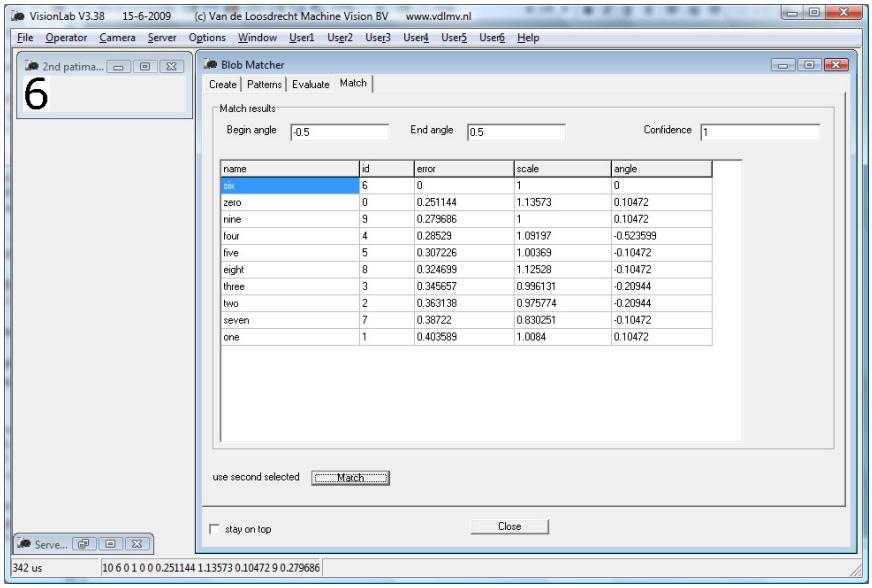
27-aug-18 Blob matching 15

Demonstration matching a blob

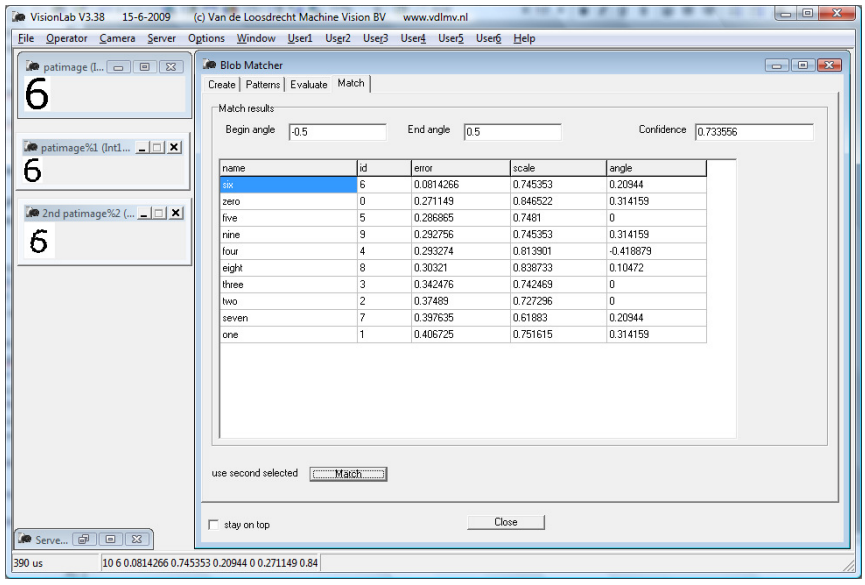
- **Extract the image for number 6 with the extract button**
- **Select image as 2nd selected**
- **Go to the Match tab**
- **Set begin angle to -0.5 and end angle to 0.5,**
  begin and end angle limit the search range for the rotation
- **Click the Match button and analyse the results**
- **Reduce the size of the image with ZoomXY 0.75 0.75 NearestPixelInterpolation**
- **RotateFull result image 0.2 0 NearestPixelInterpolation**
- **Click the Match button and analyse the results**

27-aug-18 Blob matching 16

Match the six



Match the scaled and rotated six



Interpreting the result of the match

- In the table the patterns are sorted on error value
- The scale value is the zoom factor for the pattern to match the size of the test image
- The angle value is the rotation angle in radians
- The error indicates the "distance" between the test image and the pattern after rotation and scaling
- The confidence indicates the "distance" between the best match and the second best match

Notes:

- The blob matcher will always try to find the best possible match, even if an unknown pattern is tested
- The quality of the match can be analyzed using the error value of the best match and the confidence value

Class Image Set (CIS)

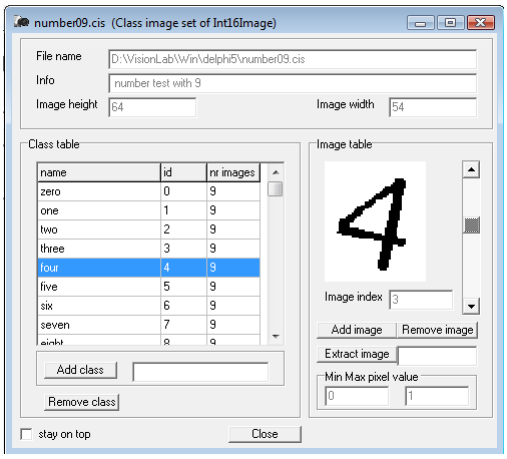
- A CIS is a collection of images with their associated classes. The image type and size are defined when the CIS is created. If both height and width are set to 0 (zero), size checking is suppressed.
- CIS:
  - Class table with the name and id of each class
  - For each class an image table,
    each image in an image table has its unique image index number
- A CIS is created in a similar way as the pattern table of a blob matcher. But there can be more then one image for each class.

For more information about CIS look in the chapter about neural networks

Demonstration class image set

- **Open CIS number09.cis**
- **Browse through class table**
- **Browse through image table**

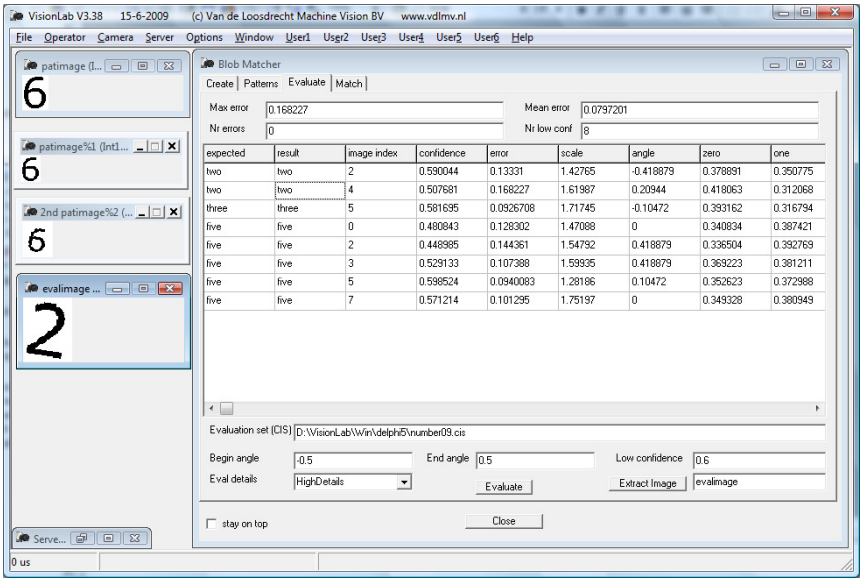
Class Image Set number09.cis



Demonstration evaluating a blob matcher

- **Go to the Evaluate tab**
- **Select the CIS number09.cis by double clicking the Evaluation set (CIS) field or typing in the correct file name**
- **Set begin angle to -0.5 and end angle to 0.5**
- **Set low confidence to 0.6**
- **Set Eval details to HighDetails**
- **Click the Evaluate button and analyze the results**
- **Note: the corresponding image of a line the result table can be inspected with double clicking on that line.**

Evaluating a blob matcher



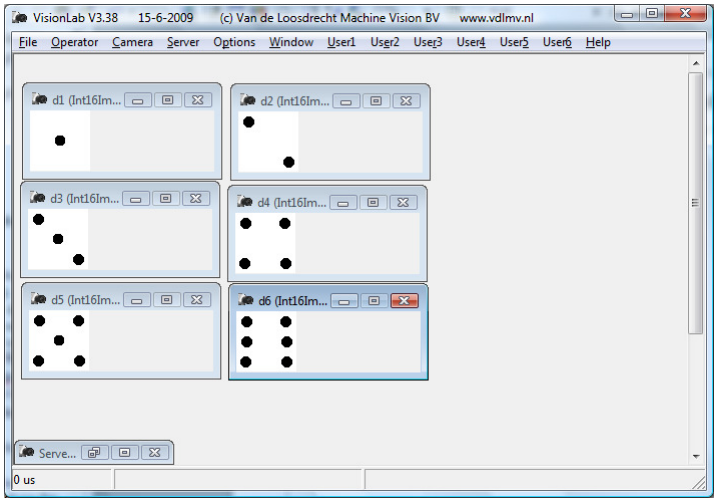
Example using blob matcher with a blob group (*)

The Blob Matcher can also be used for images that contain a group of blobs

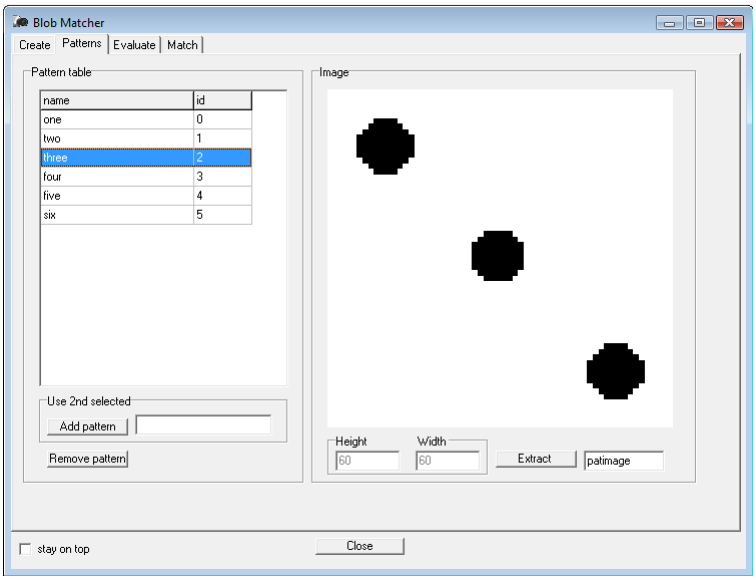
In the next example the eyes of dice will be matched

Use script BM_dice.jls to generate test images, dice.pm and evaluation dice.cis

Example using blob matcher with a blob group (*)

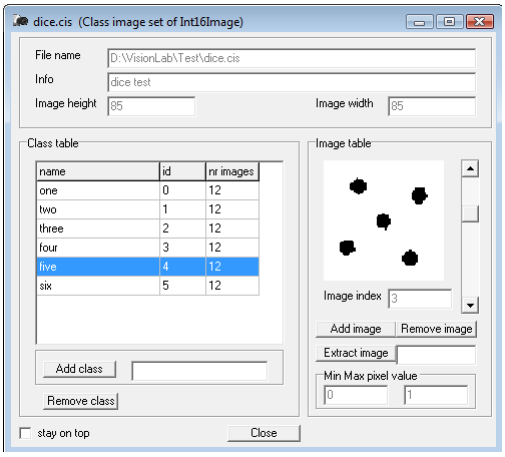


Example using blob matcher with a blob group (*)



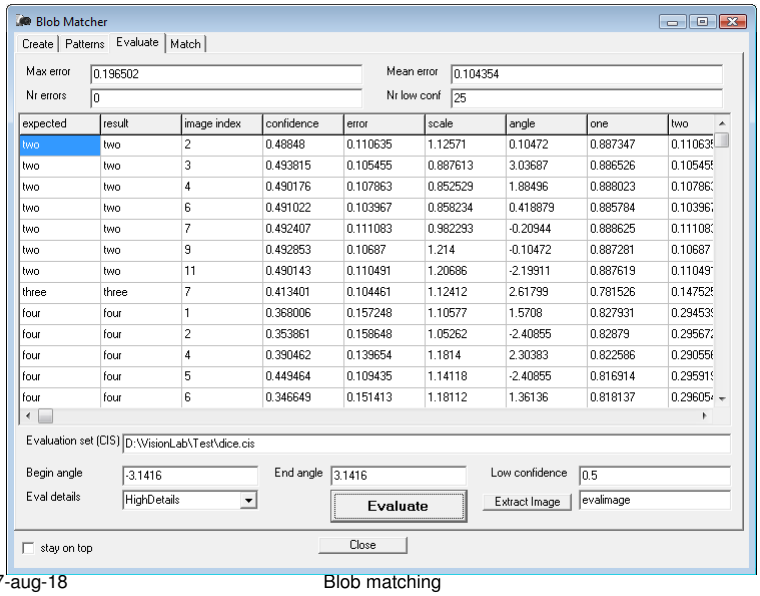
27-aug-18 Blob matching 27

Example using blob matcher with a blob group (*)



27-aug-18 Blob matching 28

Example using blob matcher with a blob group (*)



Exercise using blob matcher with a blob group (*)

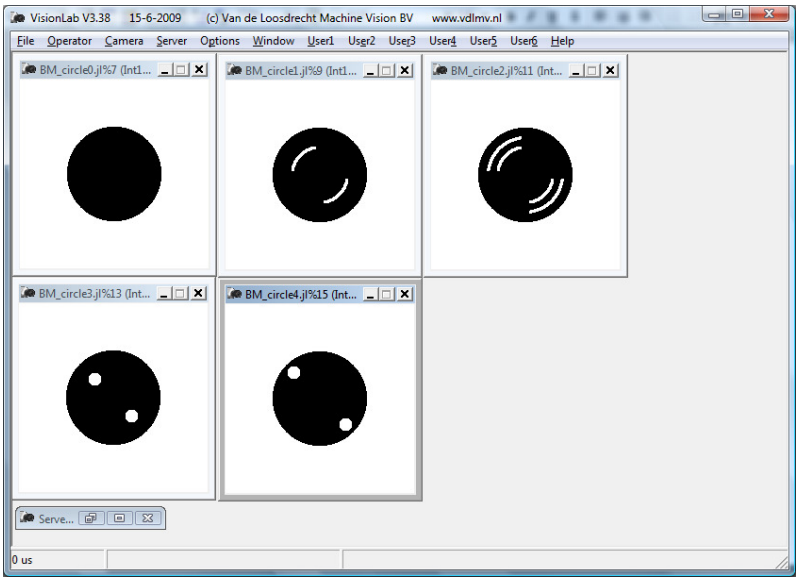
Experiment with the parameters of blob matcher in order to get less low confidences

Exercise 1 blob matcher

- **Experiment with the parameters:**
  - **number of rotations**
  - **perimeterFillRatio and fillSampleSize**
  - **(set randomSampleSize = 0)**
- **Change parameters and use CIS to evaluate, look at changes in:**
  - **Accuracy (like nr errors and nr low confidences)**
  - **Speed of execution**

27-aug-18 Blob matching 31

Exercise 2 blob matcher



27-aug-18 Blob matching 32

Exercise 2 blob matcher

- **Create a blob matcher with the 5 patterns in the image BM_circles0 .. BM_circles4**
- **Test blob matcher with different parameters on test images with different scaling and rotation**
- **Can you find a setting with parameter perimeterFillRatio = 0? Explain why**

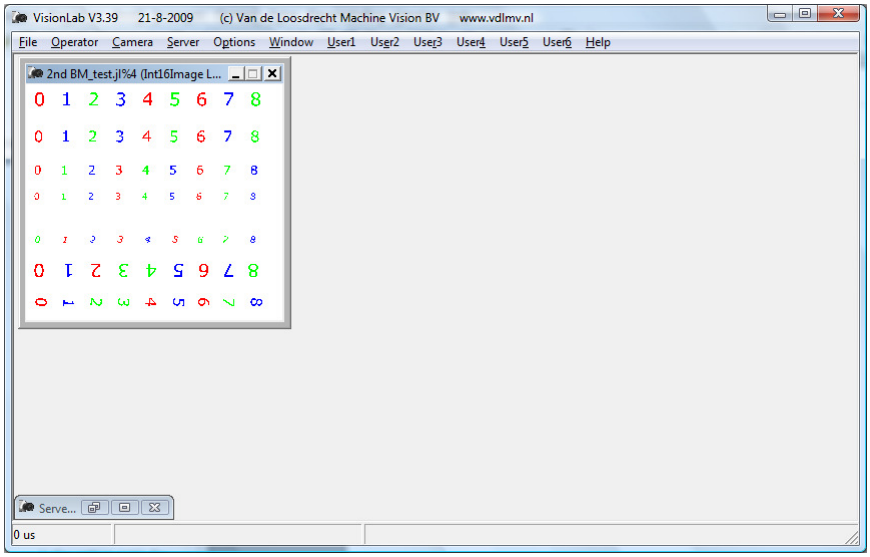
See for answer script BM_circles.jls

Demonstration finding all patterns in an image

- **NOTE: this operator works with a labeled image**
- **Open image BM_test.jl**
- **ThresholdIsoData image DarkObject**
- **Label image EightConnected**
- **Open blob matcher test08.pm**
  NrOfRatations=80, FillRatio=1, SamplesSize=20, RandomSize =0
- **Show patterns (no '9' because of rotation symmetry with '6')**
- **Select Find tab**
- **Second select labeled image using F5**
- **Find patterns with MaxError = 0.15, MinConf = 0.4, Angle <-3.14 .. 3.14>**
- **Click on entry in the label table to show position of the label in the image**
- **Click on entry in the pattern table to show all blobs classified as specified pattern in the image.**

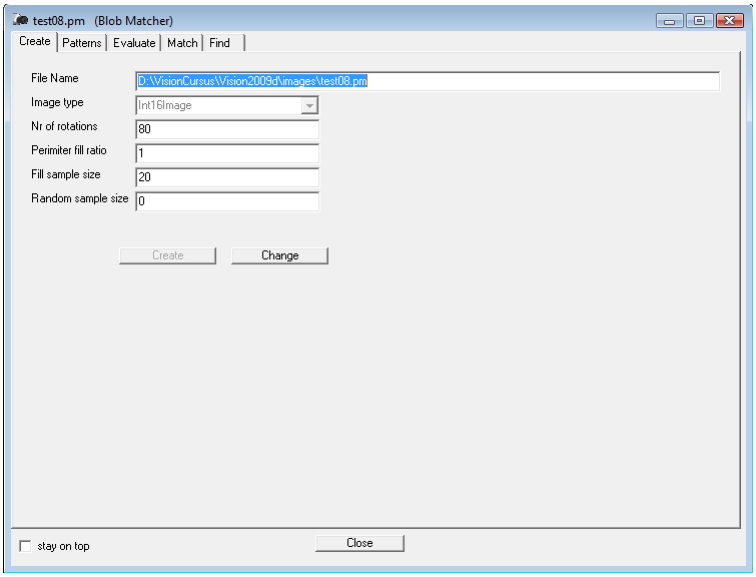
- **Note: this operator can NOT be used for blob groups**

Image BM_test.jl after labelling



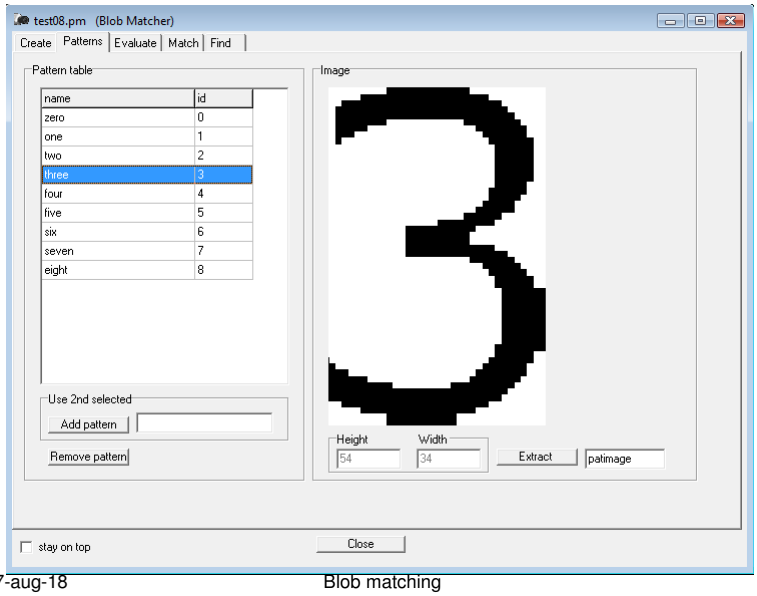
27-aug-18 Blob matching 35

Blob Matcher parameters



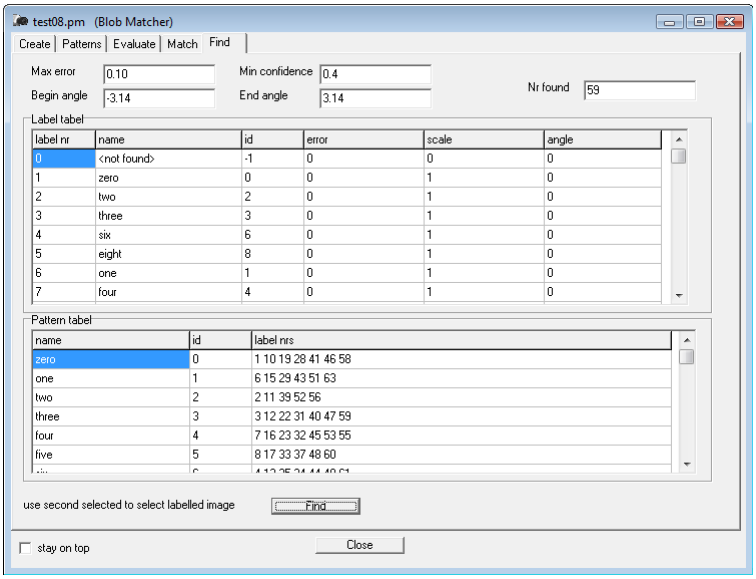
27-aug-18 Blob matching 36

Blob Matcher patterns



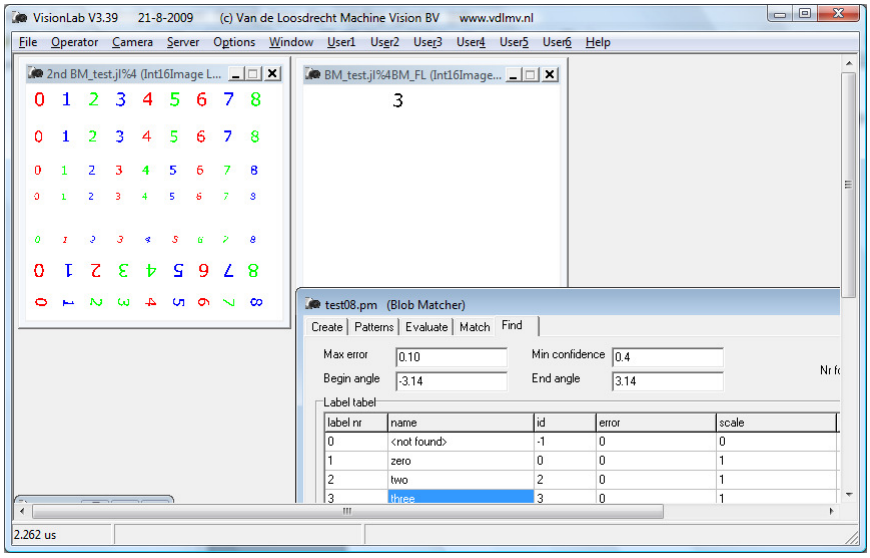
27-aug-18 Blob matching 37

The result of the search



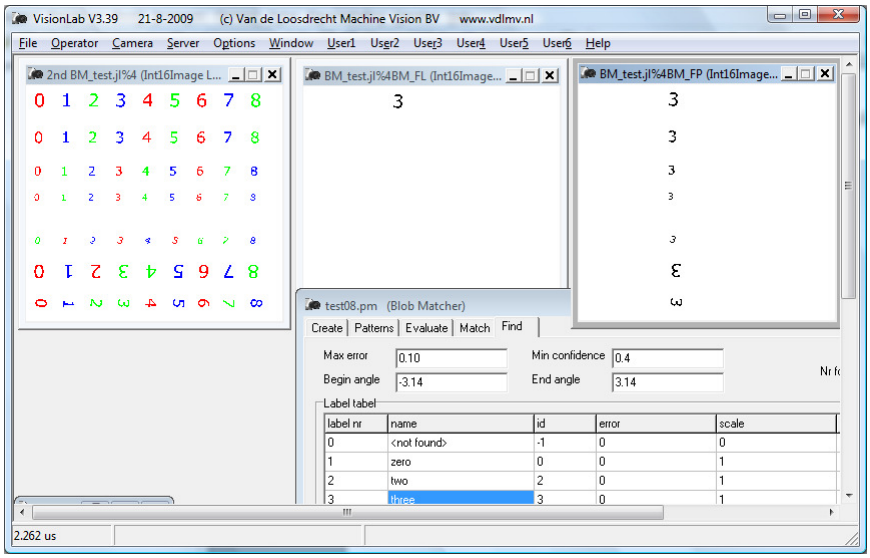
27-aug-18 Blob matching 38

Click on label number 3 in label table



27-aug-18 Blob matching 39

Click on pattern 3 in pattern table



27-aug-18 Blob matching 40

Interpreting the result of the search for all patterns

- **In the label table for each label the pattern found is enumerated together with the match error, the scale factor and rotation angle**
  If a blob in the image can not be classified with error < max error and confidence > min conf, its entry in the label table will be <not found>
  For convenience of using the operator PM_FindPatterns in scripts the first entry in the label table is for label 0 (= back ground), so label nr is index in array
- **In the pattern table for each pattern the corresponding label numbers are enumerated**
  In scripts the label nrs field can be used with ThresholdMulti to select the corresponding blobs

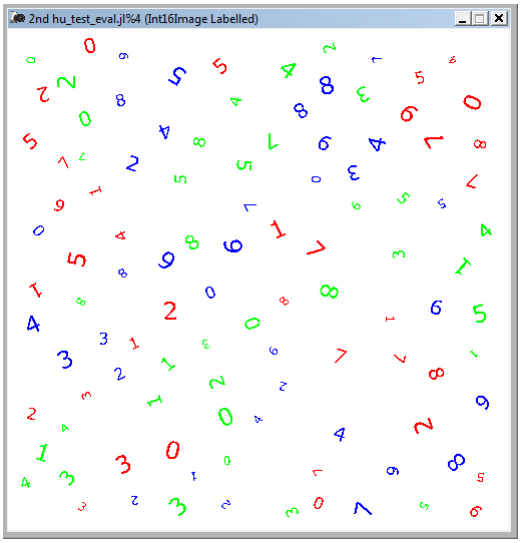
27-aug-18 Blob matching 41

Demonstration speed of FindAllPatterns

- **Open image hu_test_eval.jl**
  (1500x1500 pixels with 107 digits 0..8, random rotated and random scaled)
- **ThresholdIsoData image DarkObject or**
  ThresholdFast image 0 134 0 255 (faster)
- **Label image EightConnected**
- **Open blob matcher test08.pm**
- **Set: NrOfRatations=40, FillRatio=0, RandomSize =0**
- **Select Find tab**
- **Second select labeled image using F5**
- **Find patterns with MaxError = 0.15, MinConf = 0.3, Angle <-3.14 .. 3.14>**

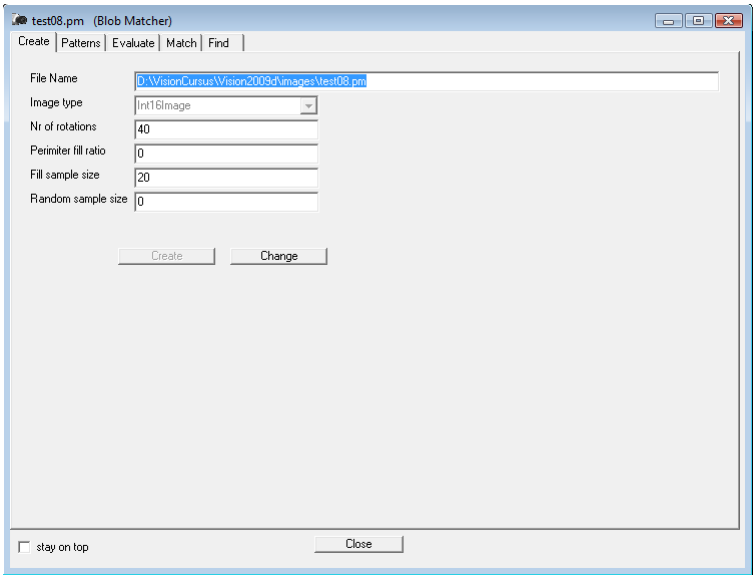
27-aug-18 Blob matching 42

Test image, 1500x1500 pixels with 107 digits 0..8, random rotated and random scaled



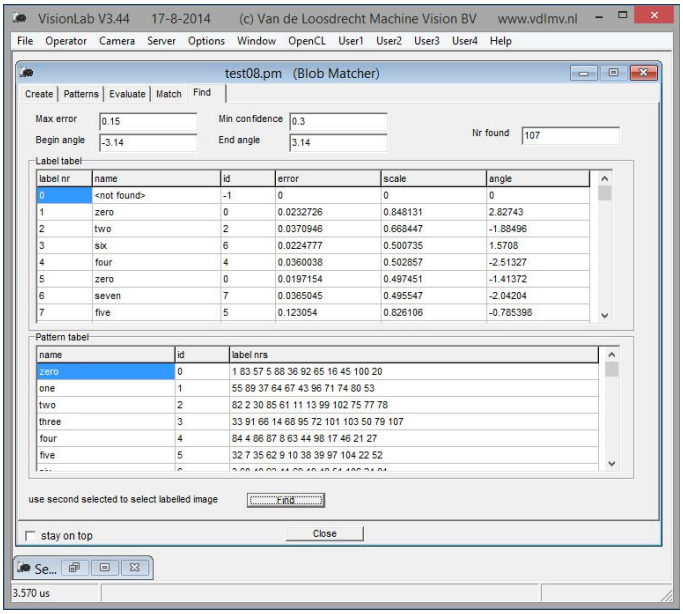
27-aug-18 Blob matching 43

Blob Matcher parameters



27-aug-18 Blob matching 44

Finding 107 random rotated and scaled numbers in 3.6 ms (Core I7 3.5 Ghz)



FindBlob

FindBlob (srcImage patImage destImage connected maxError beginAngle endAngle nrOfRotations perimeterFillRatio fillSampleSize randomSampleSize)

The FindBlob operator finds a blob in a image

All images are binary images
The result of the search is an image with the blobs found.

Menu: Operator | Pattern Matchtng

FindBlob (*)

The implementation of the FindBlob in Vision Lab uses a combination of 3 techniques:

- **Contour matching, compulsory**
  Parameter: number of rotations
  Only the contour pixels are used to find the scaling and rotation
- **"Filling" matching, optional**
  Parameters: perimeterFillRatio and fillSampleSize
  The interior of the object is sampled to find more accurate scaling and rotation
- **Random check, optional**
  Parameter: randomSampleSize
  Maximum number of pixels which are selected to check the matching

27-aug-18 Blob matching 47

FindBlob parameters (*)

***srcImage*: name of the binary image to search in**

***patImage*: name of the binary image with the pattern to search**

***destImage*: name of the binary image with the result of the search**

***connected*: the connectivity for the images, FourConnected or Eightconnected**

***beginAngle* and *endAngle*: limit the search range for the rotation**

nrOfRotations: the number of rotations in the search space

***perimeterFillRatio*: error ratio between contour match and interior match**

***fillSampleSize*: number of pixels sampled on ray in interior of blob**

***randomSampleSize*: maximum number of pixels which are selected to check the matching**

27-aug-18 Blob matching 48

Demonstration finding one blob in an image

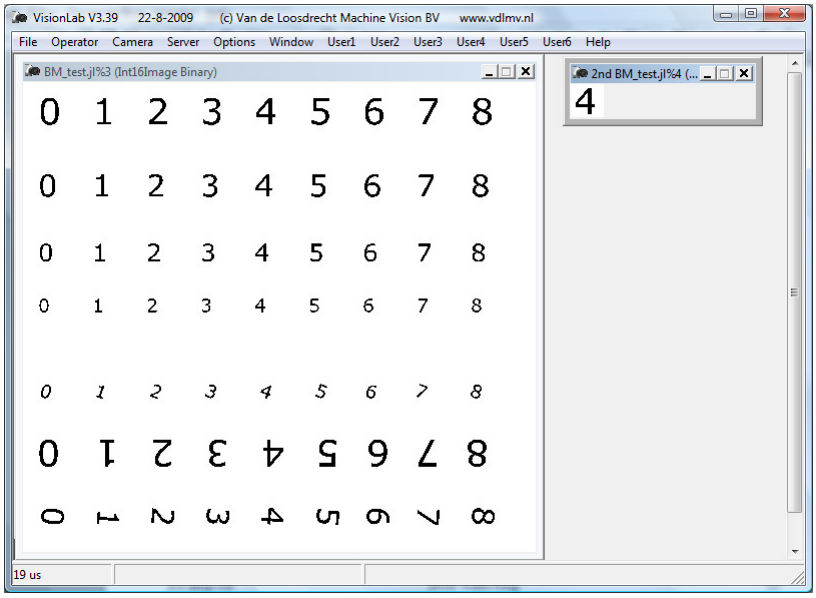
FindBlob can be used to find one single pattern in a labeled image without the need to create a complete Blob Matcher

- **Open image BM_test.jl**
- **ThresholdIsoData image DarkObject**
- **Use ROI to select a random number**
- **Second select roi image using F5**
- **Select image**
- **FindBlob on image, see parameters in slide**
- **See result of search**

- **Note: this operator can NOT be used for blob groups**

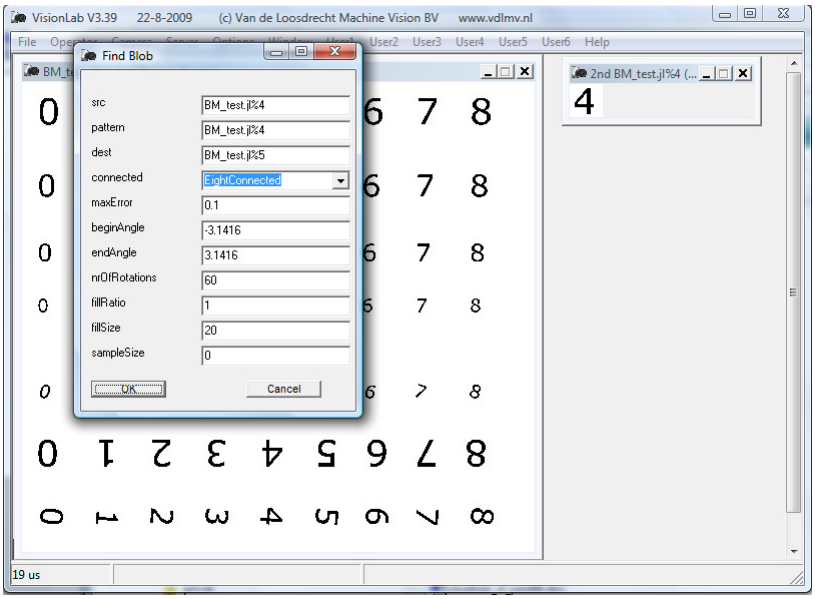
27-aug-18 Blob matching 49

Image BM_test.jl and pattern to search for



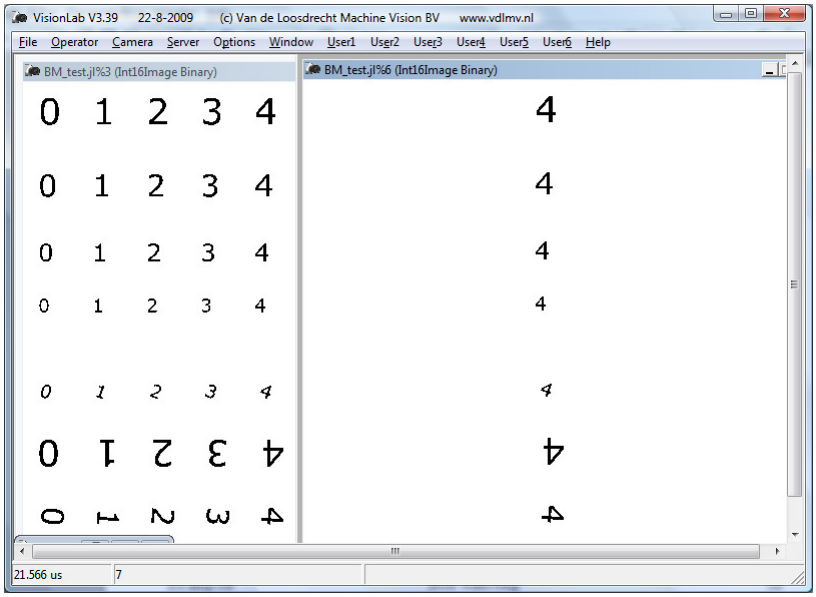
27-aug-18 Blob matching 50

FindBlob parameters



27-aug-18 Blob matching 51

Result of FindBlob



27-aug-18 Blob matching 52

Comparing with alternative methods

Alternatives for Blob Matching are:

- **Neural networks**
  - **Image classification**
  - **Feature vector classification**
- **Correlation**

Related presentations:

- **Classification with neural networks part I and part II**
  with examples how to use neural networks for classifying objects
- **Fourier Transforms**
  using correlation to match objects

27-aug-18 Blob matching 53

Comparing with alternative methods

Blob matching

Advantage:

- **Rotation and scale invariant**
- **Fast**
- **Easy training, only one template needed, few parameters**

Disadvantage:

- **Works only well on correctly segmented binary images**
- **Can not handle occlusion**

27-aug-18 Blob matching 54

Comparing with alternative methods

Neural network (image classifying)

Advantage:
- **Can work with grayscale and color images, can use grayscale or color info to discriminate**
- **Can use multiple patterns for one class in training to handle some degree of variation**

Disadvantage:
- **Complex setting of the learning parameters**
- **Can not handle scaling or rotation**

Comparing with alternative methods

Neural network (feature vector)

Advantage:
- **Can work with grayscale and color images, can use grayscale or color info to discriminate**
- **Can use multiple patterns for one class in training to handle some degree of variation**
- **Can be scaling and/or rotation invariant if features are invariant**

Disadvantage:
- **Complex setting of the learning parameters**
- **Choosing the right features can be difficult**

Comparing with alternative methods

Correlation (FFT)

Advantage:
- **Simple to use**

Disadvantage:
- **Slow**
- **Can not handle scaling or rotation**
- **Can only compare with one template at a time**

Using blob matching in scripts (*)

- **Overview script commands**
- **Details on script commands**
- **Example script BM_example.jls**
- **Example script BM_findpattern.jls**
- **Class Lexicon**

Overview script commands (1) (*)

- **PM_AddPattern <imageName> <pmName>**
- **PM_AllMatches <imageName> <pmName> <beginAngle> <endAngle> <&$array>**
- **PM_BestMatch <imageName> <pmName> <beginAngle> <endAngle>**
- **PM_ChangeBlobMatcher <bmName> <nrOfRotations> <perimeterFillRatio> <fillSampleSize> <randomSampleSize>**
- **PM_CreateBlobMatcher <bmName> <imagetype> <nrOfRotations> <perimeterFillRatio> <fillSampleSize> <randomSampleSize>**
- **PM_Delete <pmName>**
- **PM_EvaluateCIS <pmName> <cisName> <beginAngle> <endAngle> <minConfidence> <detailsStr>**
- **PM_FindPatterns <imageName> <pmName> <maxError> <minConfindence> <beginAngle> <endAngle> <&$labelTab> <&$patTab>**

27-aug-18 Blob matching 59

Overview script commands (2) (*)

- **PM_GetImageType <pmName>**
- **PM_GetParams <pmName>**
- **PM_GetPatternsNameTab <pmName>**
- **PM_PatternName <pmName> <id>**
- **PM_PatternId <pmName> <name>**
- **PM_PatternNameImage <pmName> <nameImage> <patternName>**
- **PM_PatternIdImage <pmName> <nameImage> <id>**
- **PM_Rename <pmNameOld> <pmNameNew>**
- **PM_ReadFromFile <pmName> <fileName>**
- **PM_RemovePattern <pmName> <name>**
- **PM_WriteToFile <pmName> <fileName>**

27-aug-18 Blob matching 60

Details script commands (*)

PM_AddPattern <imageName> <pmName>
Add image as pattern to pattern matcher

PM_AllMatches <imageName> <pmName> <beginAngle> <endAngle> <&$array>
Match image with all patterns and give result in an array with tuples (<patternId> <error> <scale> <angle>) sorted on lowest error

PM_BestMatch <imageName> <pmName> <beginAngle> <endAngle>
Find the best match for image with all patterns and give result in an array
Function result is <patternId> <confidence> <error> <scale> <angle>

PM_ChangeBlobMatcher <bmName> <nrOfRotations> <perimeterFillRatio> <fillSampleSize> <randomSampleSize>
Change the parameters of the blob matcher

Details script commands (*)

PM_CreateBlobMatcher <bmName> <imagetype> <nrOfRotations> <perimeterFillRatio> <fillSampleSize> <randomSampleSize>
Create a blob matcher

PM_Delete <pmName>
Delete pattern matcher from memory

PM_EvaluateCIS <pmName> <cisName> <beginAngle> <endAngle> <minConfidence> <detailsStr>
Evaluate blob matcher with a Class Image Set
Function result is string with the evaluation report

Details script commands (*)

PM_GetImageType <pmName>
Get image type of pattern matcher

PM_GetParams <pmName>
Get the parameters of the pattern matcher
Function result is a string with the parameters

PM_GetPatternsNameTab <pmName>
Function result is a string with <nr of patterns> [<patternName> <patternId>]

Details script commands (*)

PM_FindPatterns <imageName> <pmName> <maxError> <minConfindence> <beginAngle> <endAngle> <&$labelTab> <&$patTab>
Find patterns in blobs of labeled image
Function result is number of patterns found
Parameters:
imageName: name of *labeled* image
pmName: name of the pattern matcher
maxError: maximum error of best found pattern for blob
minConfindence: minimum confidence of best found pattern for blob
beginAngle and endAngle: limit the search range for the rotation
&$labelTab: an array with tupples (<patternId> <error> <scale> <angle>). Index is labelNr. A table with the search results for each label, including the BackGround (= 0). If for blob with label search was not successfull, patternId = -1 will be used
&$patternTab: an array with tupples (<labelnr1> .. <labelnrN>), index is patternId. A table with label numbers of the blobs for which the pattern was found

Details script commands (*)

PM_NumberOfPatterns <pmName>
Function result is a string with <nr of patterns>

PM_PatternName <pmName> <id>
Function result is a string with the pattern name for the specified pattern id

PM_PatternId <pmName> <name>
Function result is a string with the pattern id for the specified pattern name

PM_PatternNameImage <pmName> <nameImage> <patternName>
Make a copy of the specified pattern image

PM_PatternIdImage <pmName> <nameImage> <id>
Make a copy of the specified pattern image

27-aug-18 Blob matching 65

Details script commands (*)

PM_Rename <pmNameOld> <pmNameNew>
Rename pattern matcher

PM_ReadFromFile <pmName> <fileName>
Read pattern matcher from file

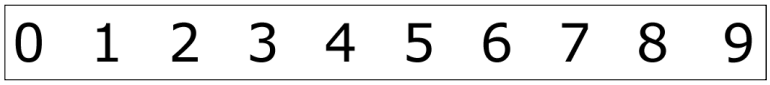
PM_RemovePattern <pmName> <name>
Remove pattern with specified name from pattern matcher

PM_WriteToFile <pmName> <fileName>
Write pattern matcher to file

27-aug-18 Blob matching 66

Example script BM_example.jls (*)

- creating a blobmacher using an image (BM_09.jl) with all patterns



- matching a test image (BM_example_test4.jl)

- result in $res:
  <patternId> <confidence> <error> <scale> <angle>

script BM_example.jls (1) (*)

```
$path = lpwd                                          Save old file path server
$oldpath = pwd
cwd $path                                             File path server = file path client

PM_CreateBlobMatcher pm Int16Image 60 1 20 0          Create Blob Matcher
$names = "zero one two three four five six seven eight nine"
VarToArray &$names &$nameTab                          Create array with the names of the
                                                      patterns
Iread allPats BM_09.jl
ThresholdIsodata allPats DarkObject                   Read image with patterns
$nrNums = LabelBlobs allPats EightConnected
$maxBlob = BlobAnalysisArray allPats &$tab SortDown TopLeft UseX Height TopLeft
    Width                                             Use BlobAnalysis to find the positions
                                                      of the patterns
```

script BM_example.jls (2) (*)

```
for $i = 0 to $maxBlob do
  VarToArray &$tab[$i] &$elm
  $label = $elm[0]
  $h = $elm[1]
  $tl = $elm[2]
  $w = $elm[3]
  $x = getnthfromvector 1 $tl
  $y = getnthfromvector 2 $tl
  ROI allPats roi $x $y $h $w
  Threshold roi $label $label
  PM_AddPattern roi pm $nameTab[$i]
endfor
PM_WriteToFile pm BM_09.pm
lread test BM_example_test4.jl
$res = PM_BestMatch test pm -0.5 0.5
PM_Delete pm
cwd $oldpath
```

For all pattern do
Find top left, width and height

Cut out the pattern
Add pattern to Blob Matcher

Write Blob Matcher to file
Test Blob Matcher with test image

Set file path server to original path

27-aug-18 Blob matching 69

Example script BM_findpattern.jls (*)

Example how to use the FindAllPatterns operator on a labeled image

- **Open BlobMatcher**
- **Open image, threshold and label**
- **FindAllPatterns**
- **Extract the fours and fives**

27-aug-18 Blob matching 70

script BM_findpatterns.jls (*)

```
$path = lpwd
$oldpath = pwd
cwd $path

PM_ReadFromFile pm test08.pm
IRead test BM_test.jl
ThresholdIsodata test DarkObject
LabelBlobs test EightConnected
PM_FindPatterns test pm 0.10 0.4 -3.14 3.14 &$labelTab &$patTab
Copy test fours
$patId = PM_PatternId pm four
ThresholdMulti fours $patTab[$patId]
Display fours
Copy test fives
$patId = PM_PatternId pm five
ThresholdMulti fives $patTab[$patId]
Display fives

PM_Delete pm
cwd $oldpath
```

Save old file path server

File path server = file path client

Read BlobMatcher from file

Read test image, threshold and label

Ask for patternId for pattern 'four'

In $patTab[$patId] are all labels for found objects (= 'four)

Set file path server to original path

27-aug-18 Blob matching 71

Class Lexicon (*)

Menu: Operator | Pattern Lexicon

A class lexicon can be to improve the reliability of classifying items that consist of more then one objects.

Example: identifying street names on street name signposts in a city.

First a classifier like a pattern matcher is trained for the character font.

Then the lexicon is filled with all street names in the city.

After the pattern matcher has identified the best match for each individual character found in the signpost, the lexicon is consulted for the best match for the street name.

In this way errors in individual matches of characters can be compensated.

27-aug-18 Blob matching 72

Class lexicon commands (*)

CLX_AddWord <clxName> <$word>
Add word to class lexicon
clxName: name of the class lexicon
word: variable with string containing all classNames of word seperated bij spaces

CLX_Clear <clxName>
Remove all words from class lexicon
clxName: name of the class lexicon

CLX_CreateClassLexicon <clxName> <$classTab>
Create class lexicon from classTab, this information can be retrived with PM_GetPatternsNameTab, CIS_GetClassTab or CFS_GetClassTab, format of classTab: <nrClasses> [<className> <classId>]
clxName: name of the class lexicon

27-aug-18 Blob matching 73

Class lexicon commands (*)

CLX_Delete <clxName>
Delete class lexicon with name clxName
clxName: name of the class lexicon

CLX_FindBestWord <clxName> <&$wordMatchTab> <&$bestWord> <optimize>
Find best matched word in class lexicon
clxName: name of the class lexicon
wordMatchTab: array with for each letter the pattern match result for all classes, format [<classId> <value>]
bestWord: string with best matched word in class lexicon, classNames are seperated by spaces
optimize: possible values = OptimizeForMinimum and OptimizeForMaximum, indicates whether FindBestWord operator searches for best minimal or maximal values

CLX_GetWords <clxName> <&$wordTab>
Get all words in class lexicon
clxName: name of the class lexicon
wordTab: array with the words
Function result is number of words in lexicon

27-aug-18 Blob matching 74

Class lexicon commands (*)

CLX_NrClasses <clxName>
Get number of classes in class lexicon
clxName: name of the class lexicon
function result is number of classes in lexicon

CLX_NrWords <clxName>
Get number of words in class lexicon
clxName: name of the class lexicon
function result is number of words in lexicon

CLX_RemoveWord <clxName> <$word>
Remove word from class lexicon
clxName: name of the class lexicon
word: variable with string containing all classNames of word seperated bij spaces

CLX_Rename <clxNameOld> <clxNameNew>
Change name of class lexicon
clxNameOld: old name of the class lexicon
clxNameNew: new name of the class lexicon

Class lexicon commands (*)

CLX_ReadFromFile <clxName> <fileName>
Read class lexicon from file
clxName: name of the class lexicon
fileName: file name to read from

CLX_WriteToFile <clxName> <fileName>
Write class lexicon to file
clxName: name of the class lexicon
fileName: file name to write to

Class lexicon commands (*)

Note: Raw class lexicons contain only classIds and no classNames

CLX_AddRawWord <clxName> <$word>
Add raw word to class lexicon
clxName: name of the class lexicon
word: variable with string containing all classIds of word seperated bij spaces

CLX_CreateRawClassLexicon <clxName> <nrClasses>
Create raw class lexicon with nrClasses classes.
clxName: name of the class lexicon

CLX_FindBestRawWord <clxName> <&$wordMatchTab> <&$bestWord> <optimize>
Find best matched word in raw class lexicon
clxName: name of the class lexicon
wordMatchTab: array with for each letter the pattern match result for all classes, format [<classId> <error>]
bestWord: string with best matched word in class lexicon, classIds are seperated by spaces
optimize: possible values = OptimizeForMinimum and OptimizeForMaximum, indicates whether FindBestWord operator searches for best minimal or maximal values

27-aug-18 Blob matching 77

Class lexicon commands (*)

CLX_RemoveRawWord <clxName> <$word>
Remove raw word from class lexicon
clxName: name of the class lexicon
word: variable with string containing all classIds of word seperated bij spaces

27-aug-18 Blob matching 78