SPAWAR Systems Center PACIFIC

Introduction to Blobs and using the OpenCV Blobs Library

Mr. Bryan Bagnall
SPAWAR Systems Center, Pacific
Phone: 619-553-4061
Email: bryan.bagnall@navy.mil

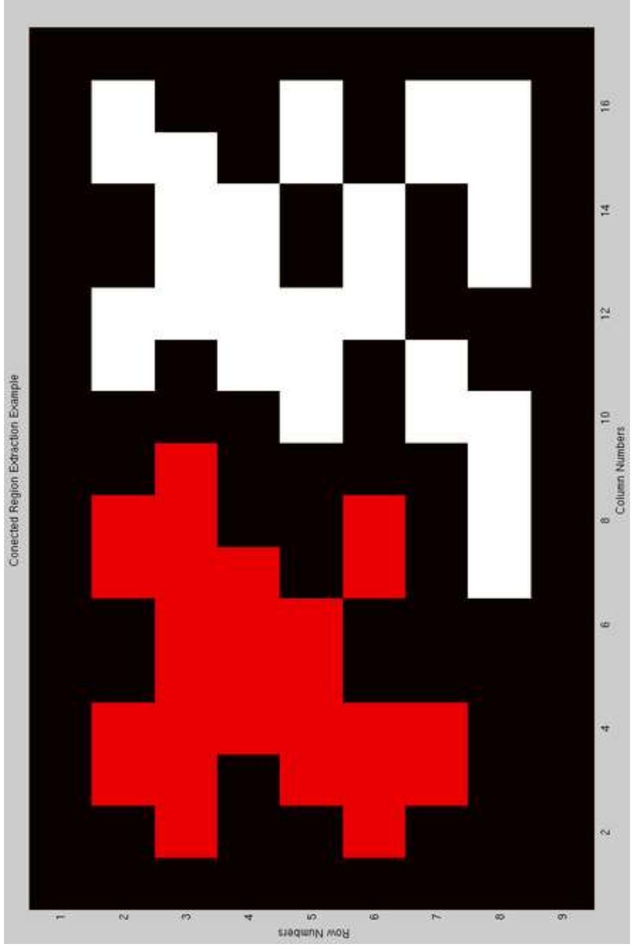
Mr. Sparta Cheung
SPAWAR Systems Center, Pacific
Phone: 619-553-5927
Email: sparta.cheung@navy.mil

Overview of Talk

- Overview of Blob extraction
- Introduction to Blob Library
- Example: Using Blobs
- Conclusions

Overview of Blob Extraction

- Identify regions in an image that differ in properties like color or brightness
- Detect connected regions in binary images
  - Usually done after thresholding
- Many different and more complex algorithms
  - We're going to discuss the two-pass method

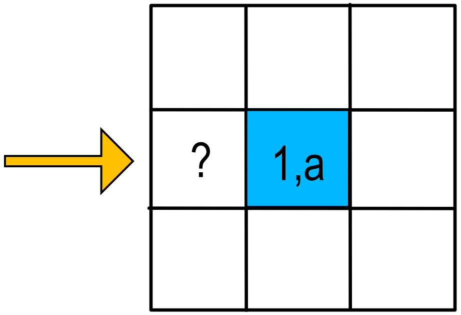


Blob Extraction – Two-pass algorithm

- Relatively easy to understand and implement
- First pass records equivalences and assigns temporary labels
- Second pass replaces temporary labels with the label of its equivalence class
- Connectivity checks are carried out by checking the labels of pixels that are North-East, North, North-West, and West of the current pixel
- Four conditions

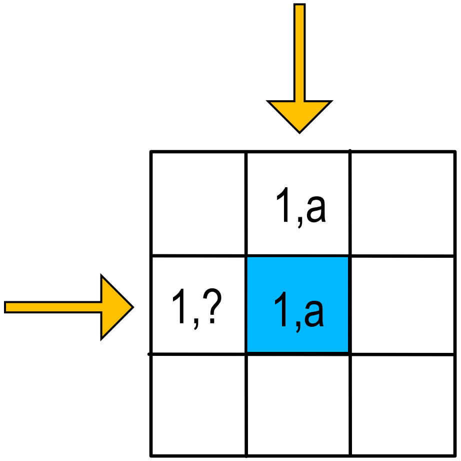
Blob Extraction – Two-pass algorithm

- 1st condition – Does the pixel to the left (west) have the same value?
  - Yes → Same region, assign the same label to the current pixel
  - No → Check next condition



?	1,a	

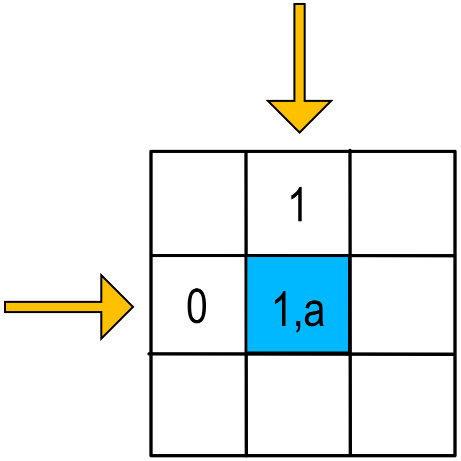
- 2nd condition – Do the pixels to the north and west of the current pixel have the same value but not the same label?
  - Yes → We know that the north and West pixels belong to the same region and must be merged, assign the current pixel the minimum of the north and west labels and record their equivalence relationship
  - No → Check next condition



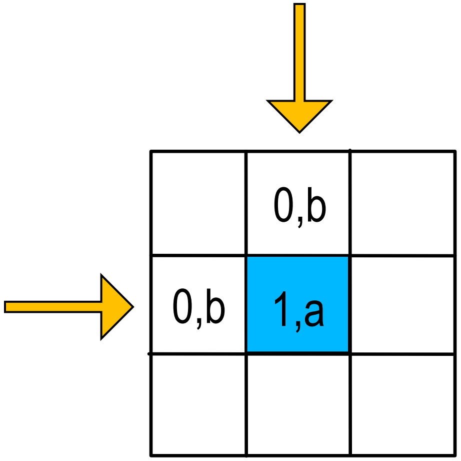
	1,a	
1,?	1,a	

Blob Extraction – Two-pass algorithm

- 3rd condition – Does the pixel to the left (west) have a different value and the one to the north the same value?
  - Yes ⟶ Assign the label of the north pixel to the current pixel
  - No ⟶ Check next condition

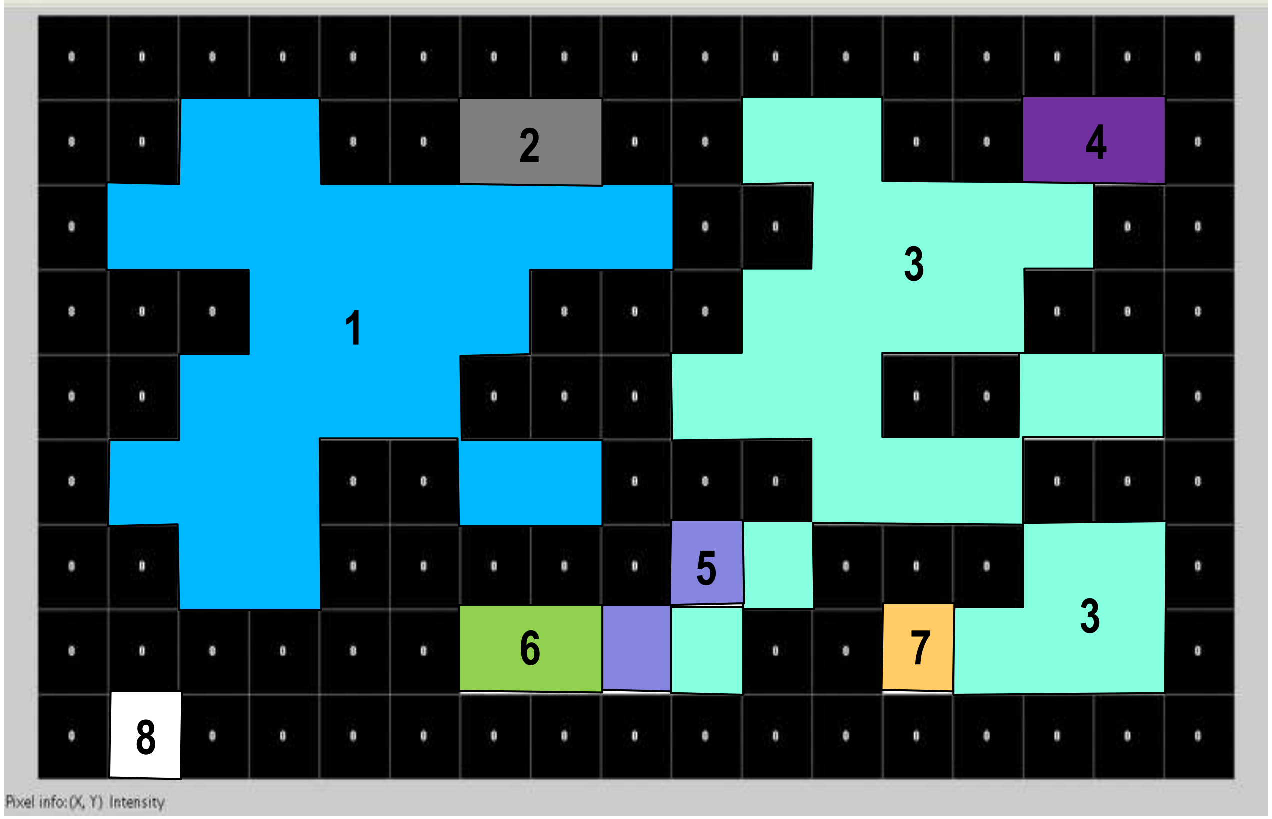


- 4th condition – Do the pixel's north and west neighbors have different pixel values?
  - Yes ⟶ Create a new label id and assign it to the current pixel



Blob Extraction – Two-pass algorithm

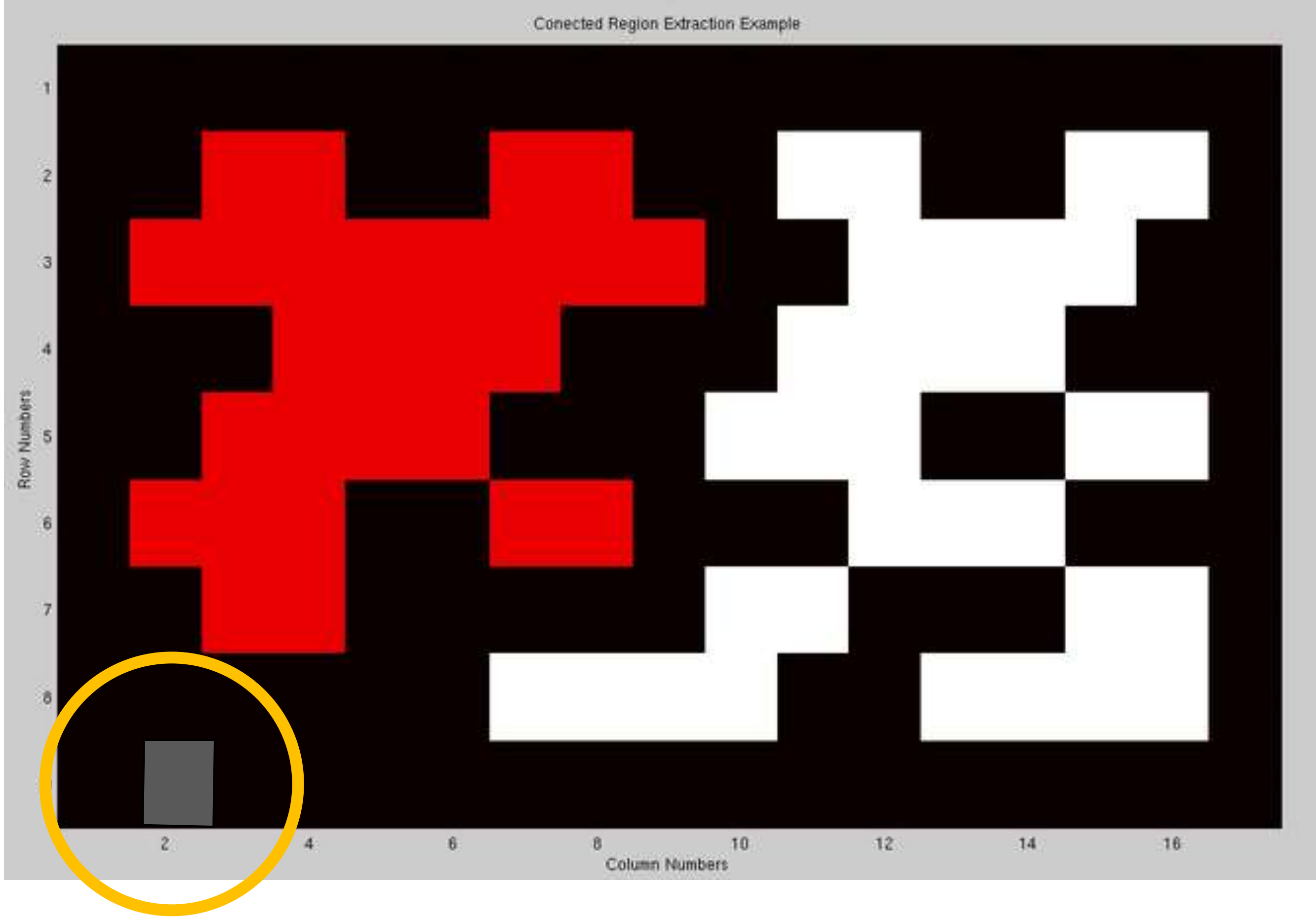
- After the first pass labels are generated for each of the pixels according to the four conditions and the equivalence tables are generated



Set ID	Equivalent Labels
1	1,2
2	1,2
3	3,4,5,6,7
4	3,4,5,6,7
5	3,4,5,6,7
6	3,4,5,6,7
7	3,4,5,6,7

Blob Extraction – Two-pass algorithm

- On the second pass the labels are merged



Blob Extraction – Two-pass algorithm

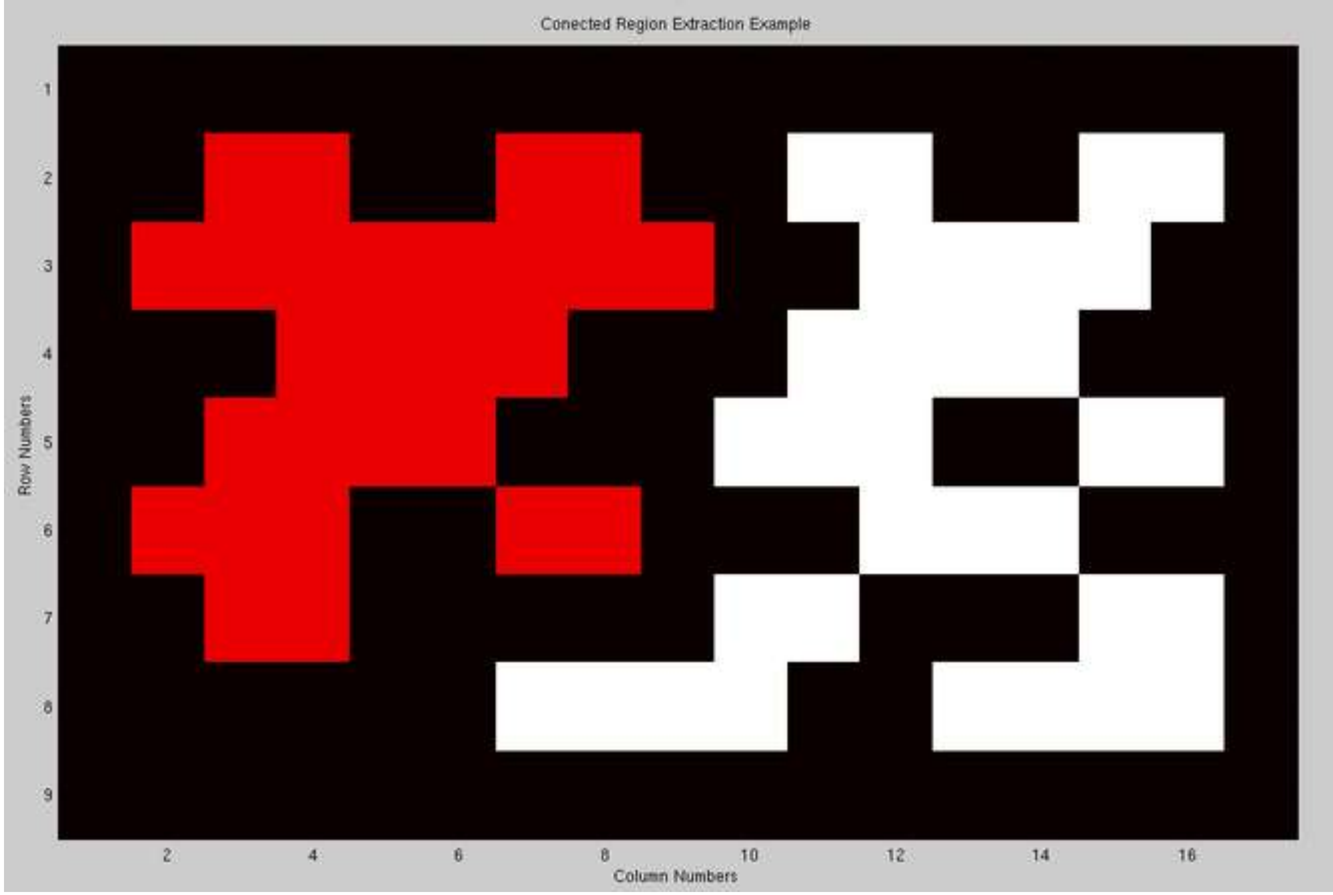
- How do we remove small specks and keep the blobs?
- ..\OpenCVThresholdAndExtractBlobs.cpp

```
blobs.Filter(blobs, B_INCLUDE, CBlobGetArea(),  B_GREATER, 100);
blobs.Filter(blobs, B_INCLUDE, CBlobGetMean(),  B_GREATER, 1);
```

- This allows us to decide how “big” a significant blob is

Blob Extraction – Two-pass algorithm

- Now only large blobs remain



Blob Extraction – Two-pass algorithm

- 1st condition – Does the pixel to the left (west) have the same value?
    - Yes → Same region, assign the same label to the current pixel
    - No → Check next condition

- 2nd condition – Do the pixels to the north and west of the current pixel have the same value but not the same label?
    - Yes → We know that the north and West pixels belong to the same region and must be merged, assign the current pixel the minimum of the north and west labels and record their equivalence relationship
    - No → Check next condition

- 3rd condition – Does the pixel to the left (west) have a different value and the one to the north the same value?
    - Yes → Assign the label of the north pixel to the current pixel
    - No → Check next condition

- 4th condition – Do the pixel's north and west neighbors have different pixel values?
    - Yes → Create a new label id and assign it to the current pixel

Introduction to Blobs Library for OpenCV

- Blobs Library is a plugin for OpenCV.
    - http://opencv.willowgarage.com/wiki/cvBlobsLib
- Replicates functionality of Matlab’s BWLabel and regionprops functions
- Takes binary image, and finds all of the blobs.
    - Blob is a “connected object”
- For each blob, finds, length, width, and orientation
- Can extract the sub-image corresponding to the blob area
- Can filter out blobs below or above a certain
    1. Area
    2. Mean value

Example: Using the blobs library with OpenCV

- Now we'll go through an example using OpenCV and the blobs library.

- The code is in the folder:
  ..\03_OpenCV_and_blobs\examples\Blobs_examples
  \OpenCVThresholdAndExtractBlobs

Example: Using the blobs library with OpenCV

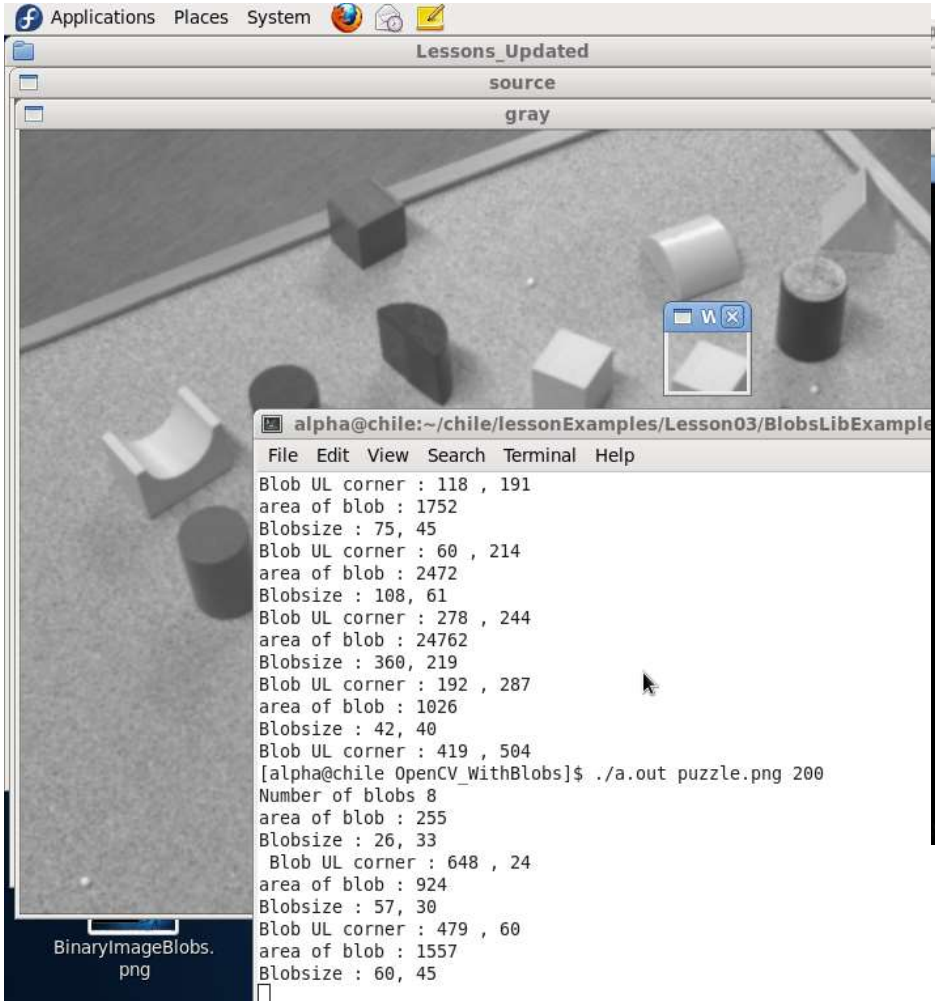
- Modify the *.bat files so that the paths are correct
  build_cmake_project_vs10.bat
  cmake_arguments.bat

- Run build_cmake_project_vs10.bat and build the project

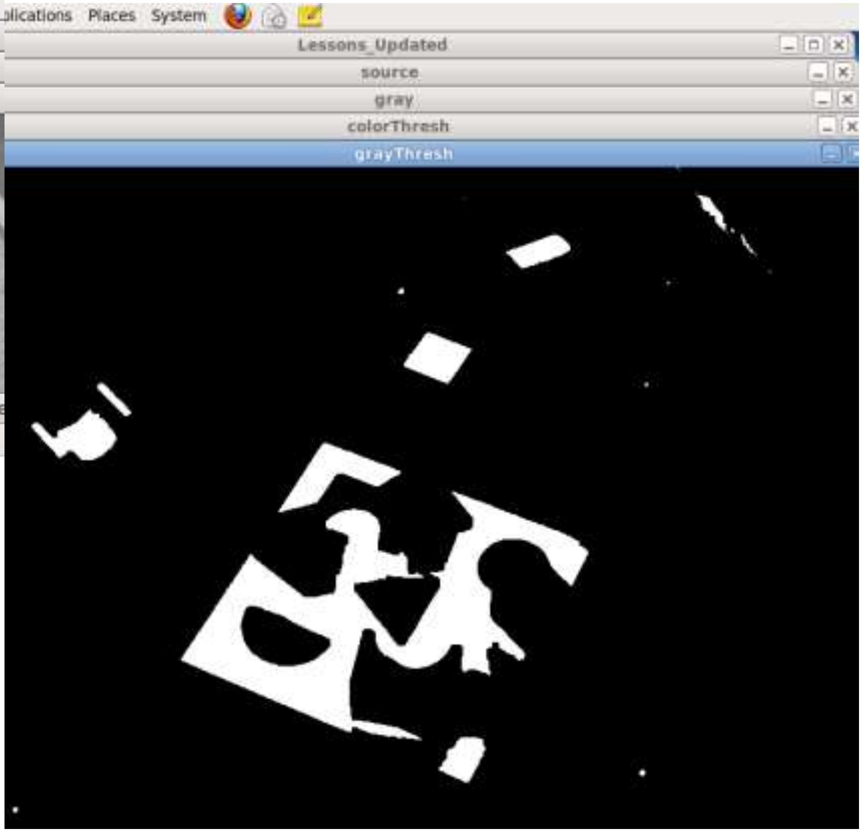
- Compile the project in Visual C++

- Run ..\Debug\runProject.bat

Blob Example Results



```
Blob UL corner : 118 , 191
area of blob : 1752
Blobsize : 75, 45
Blob UL corner : 60 , 214
area of blob : 2472
Blobsize : 108, 61
Blob UL corner : 278 , 244
area of blob : 24762
Blobsize : 360, 219
Blob UL corner : 192 , 287
area of blob : 1026
Blobsize : 42, 40
Blob UL corner : 419 , 504
[alpha@chile OpenCV_WithBlobs]$ ./a.out puzzle.png 200
Number of blobs 8
area of blob : 255
Blobsize : 26, 33
 Blob UL corner : 648 , 24
area of blob : 924
Blobsize : 57, 30
Blob UL corner : 479 , 60
area of blob : 1557
Blobsize : 60, 45
```



Another Sample Image

This simple algorithm can work with some “easy” images