Introduction to Blobs and using the OpenCV Blobs Library

Mr. John Stastny
SPAWAR Systems Center, Pacific

Mr. Bryan Bagnall
SPAWAR Systems Center, Pacific

Mr. Lucas Keenan
SPAWAR Systems Center, Pacific

Disclaimer

▼ Though SSC Pacific makes every effort to perform quality assurance on its training materials, the material in this presentation may inadvertently include technical inaccuracies or other errors. We would be grateful if users notify us of any errors or inaccuracies they may find.

▼ The presentation contains references to links and to third-party websites. These are provided for the convenience and interest of users and this implies neither responsibility for, nor approval of, information contained in these websites on the part of the U.S. Government. The USG makes no warranty, either express or implied, as to the accuracy, availability or content of information, text, graphics in the links/third party websites. The USG has not tested any software located at these sites and does not make any representation as to the quality, safety, reliability or suitability of such software, nor does this presentation serve to endorse the use of such sites.

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
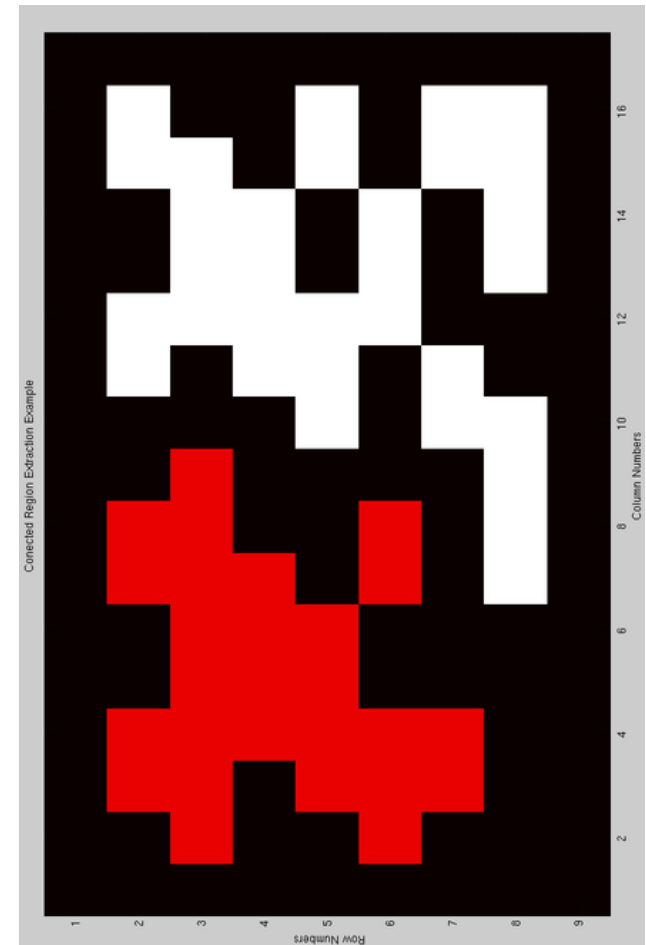
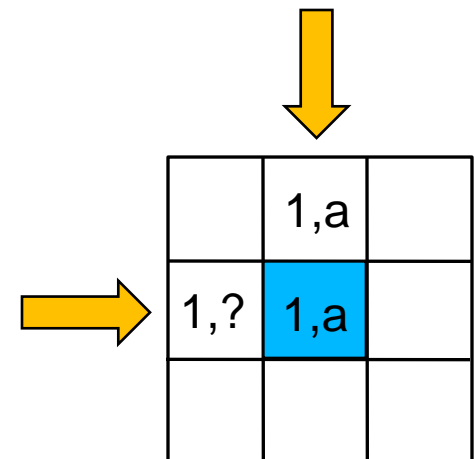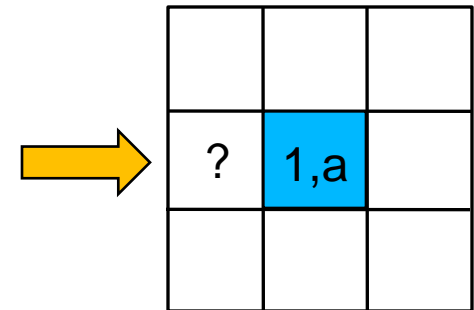# Blob Extraction – Two-pass algorithm

▼ 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
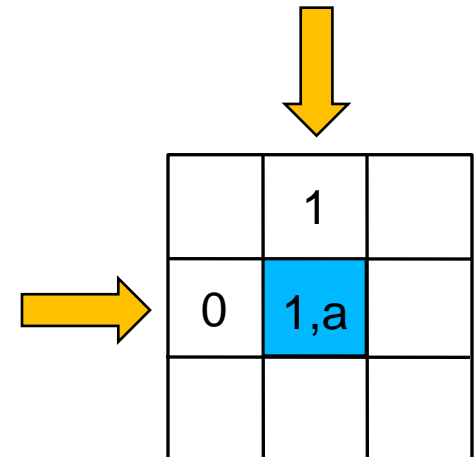
▼ Four conditions

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

▼ 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
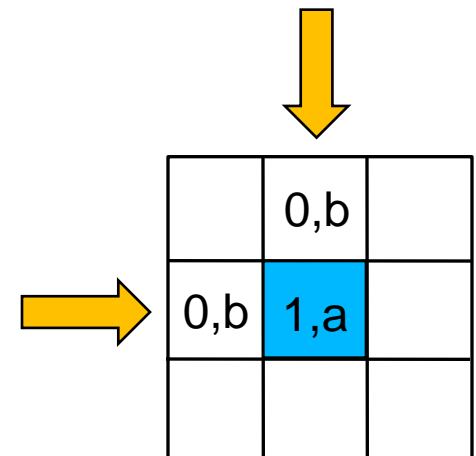- ▼ No → Check next condition

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
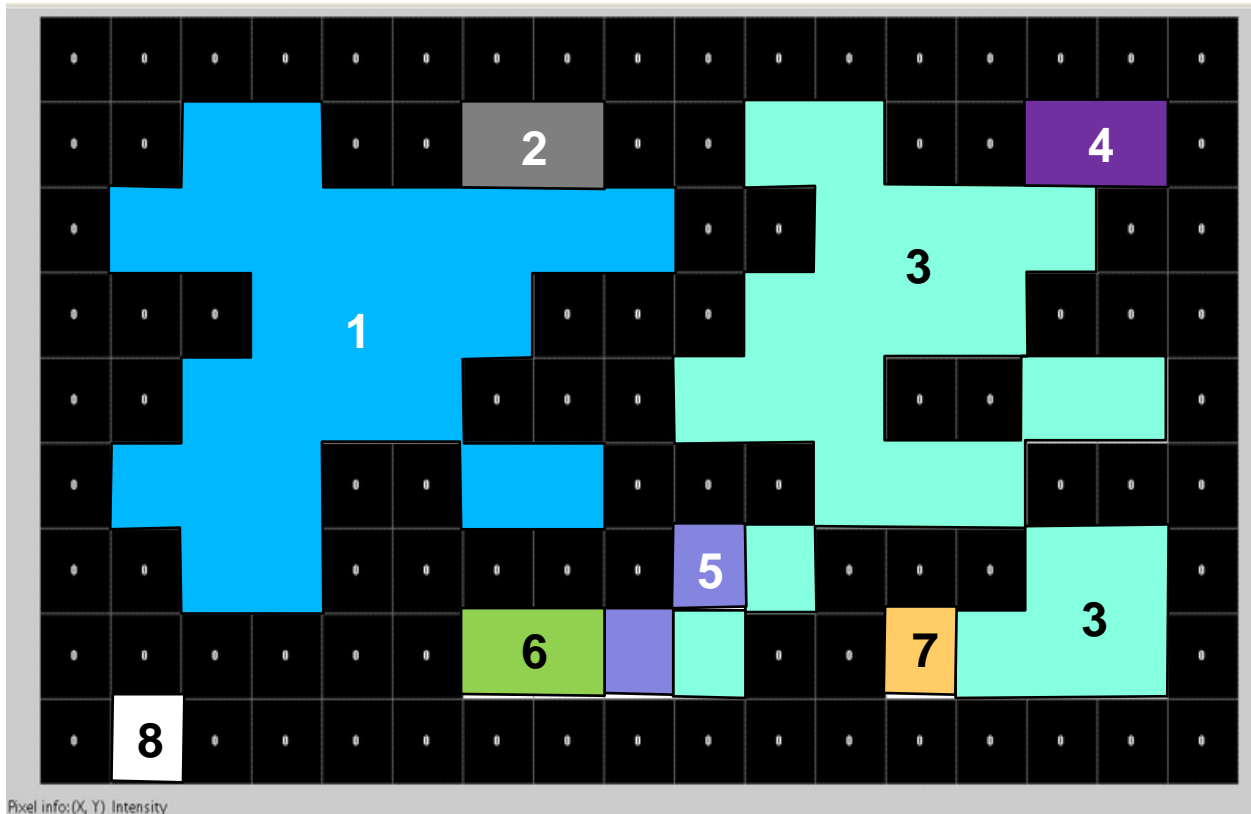
	1	
0	1,a	

▼ 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

|	0,b	
0,b	1,a	

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

Pixel info:(X, Y) Intensity

Blob Extraction – Two-pass algorithm

▼ On the second pass the labels are merged

Conected Region Extraction Example
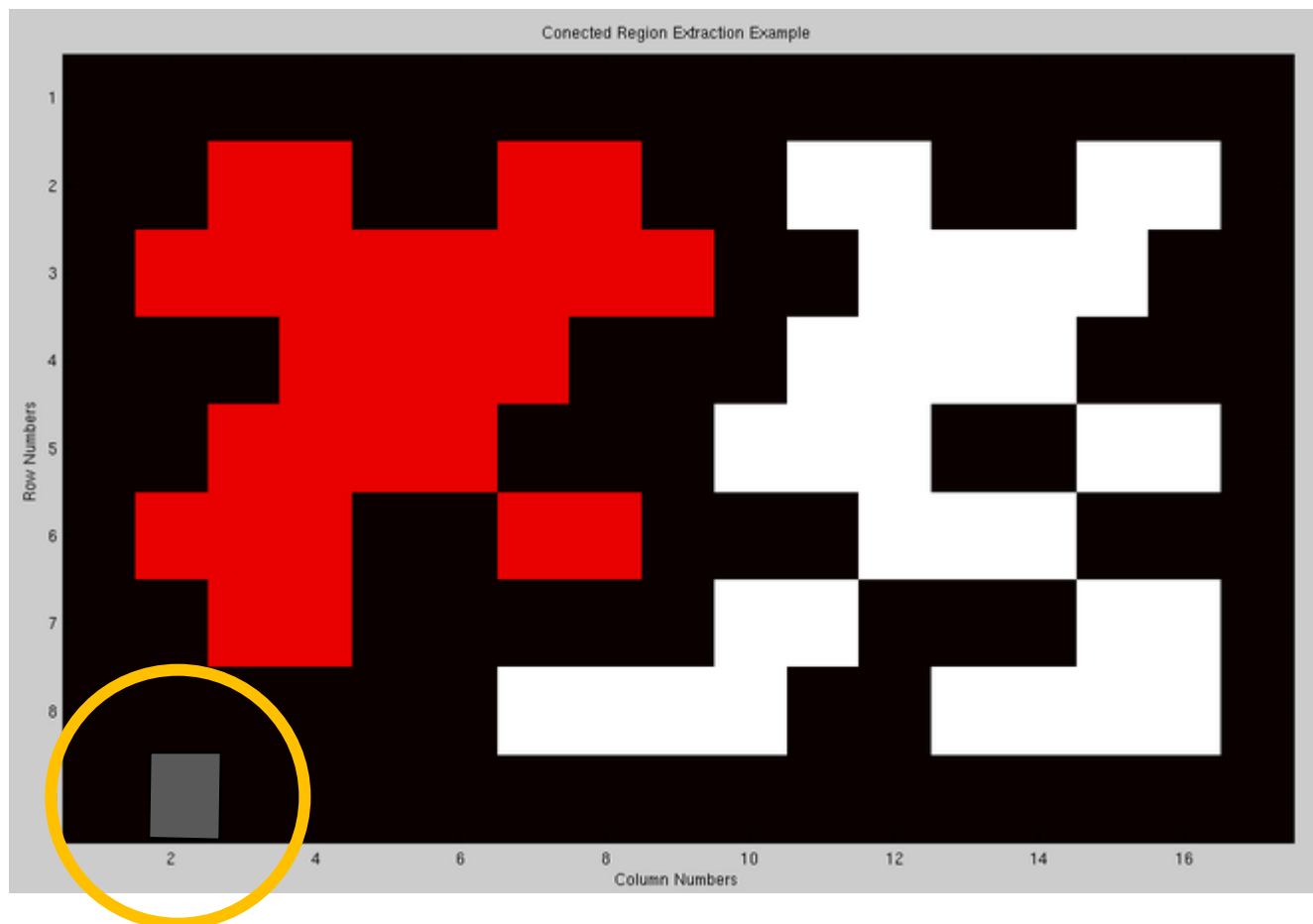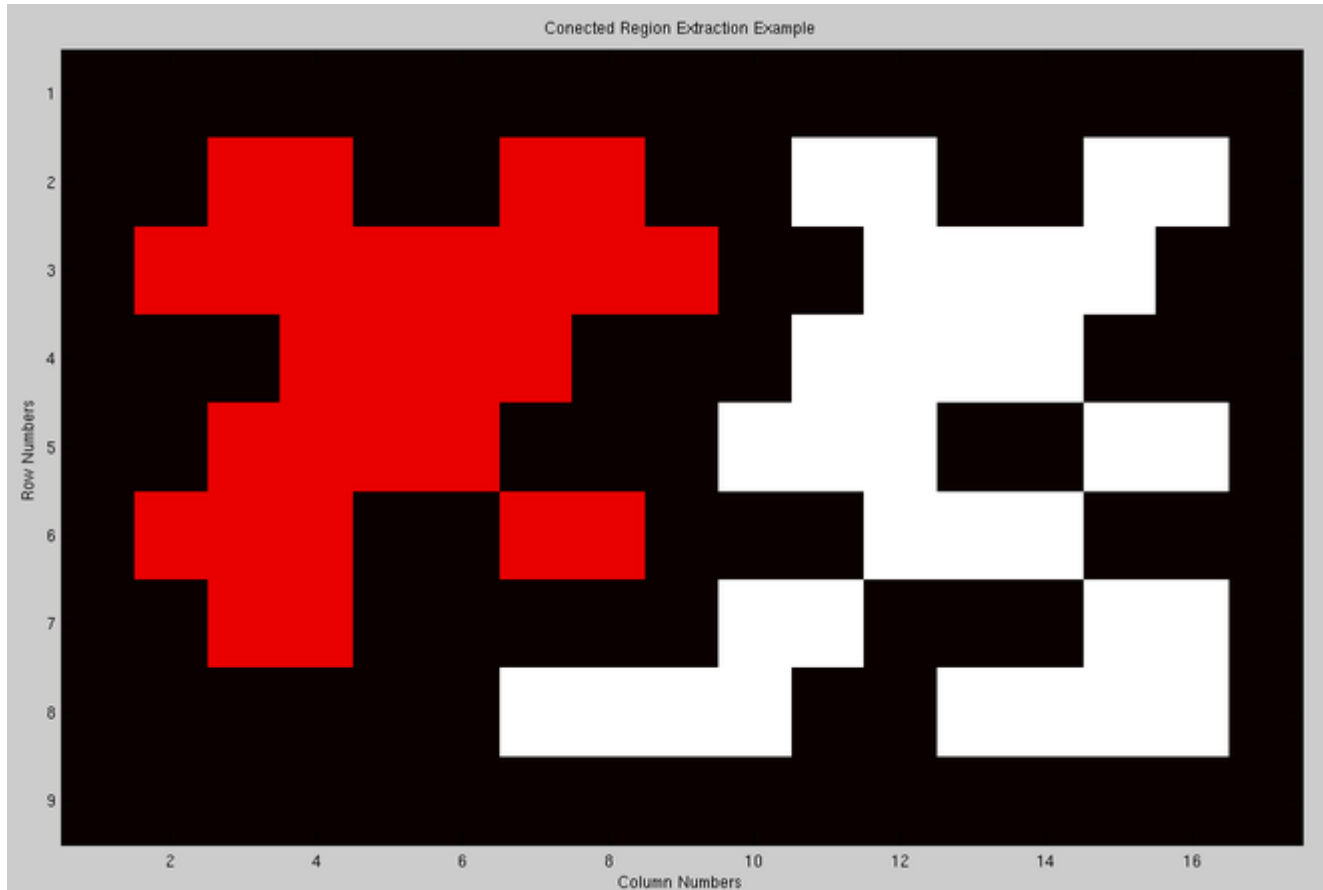
Blob Extraction – Two-pass algorithm

▼   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

▼ Now only large blobs remain

Conected Region Extraction Example

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:
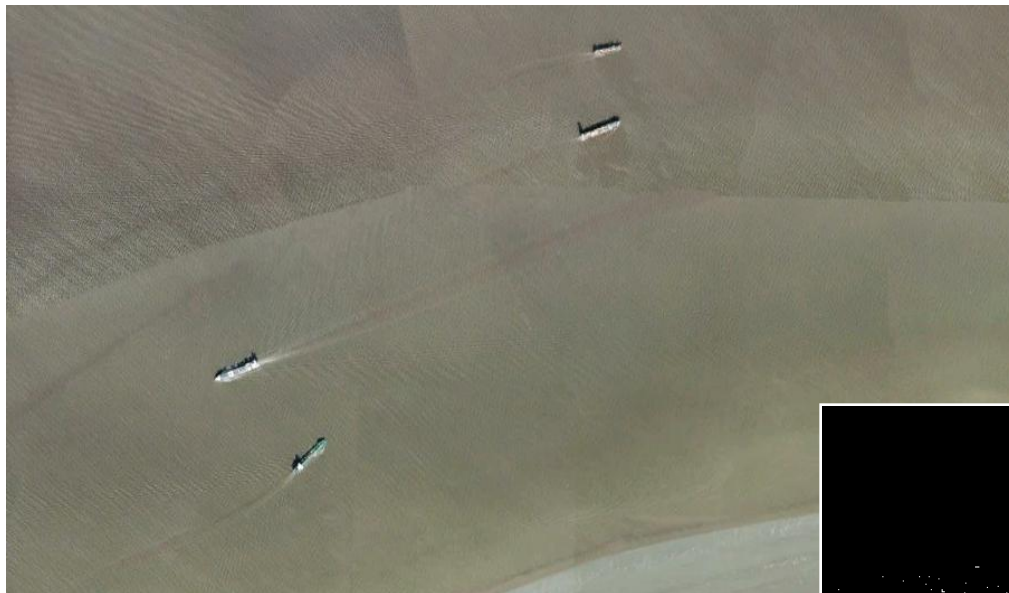..\03_OpenCV_and_blobs\examples\Blobs_examples
\OpenCVThresholdAndExtractBlobs

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
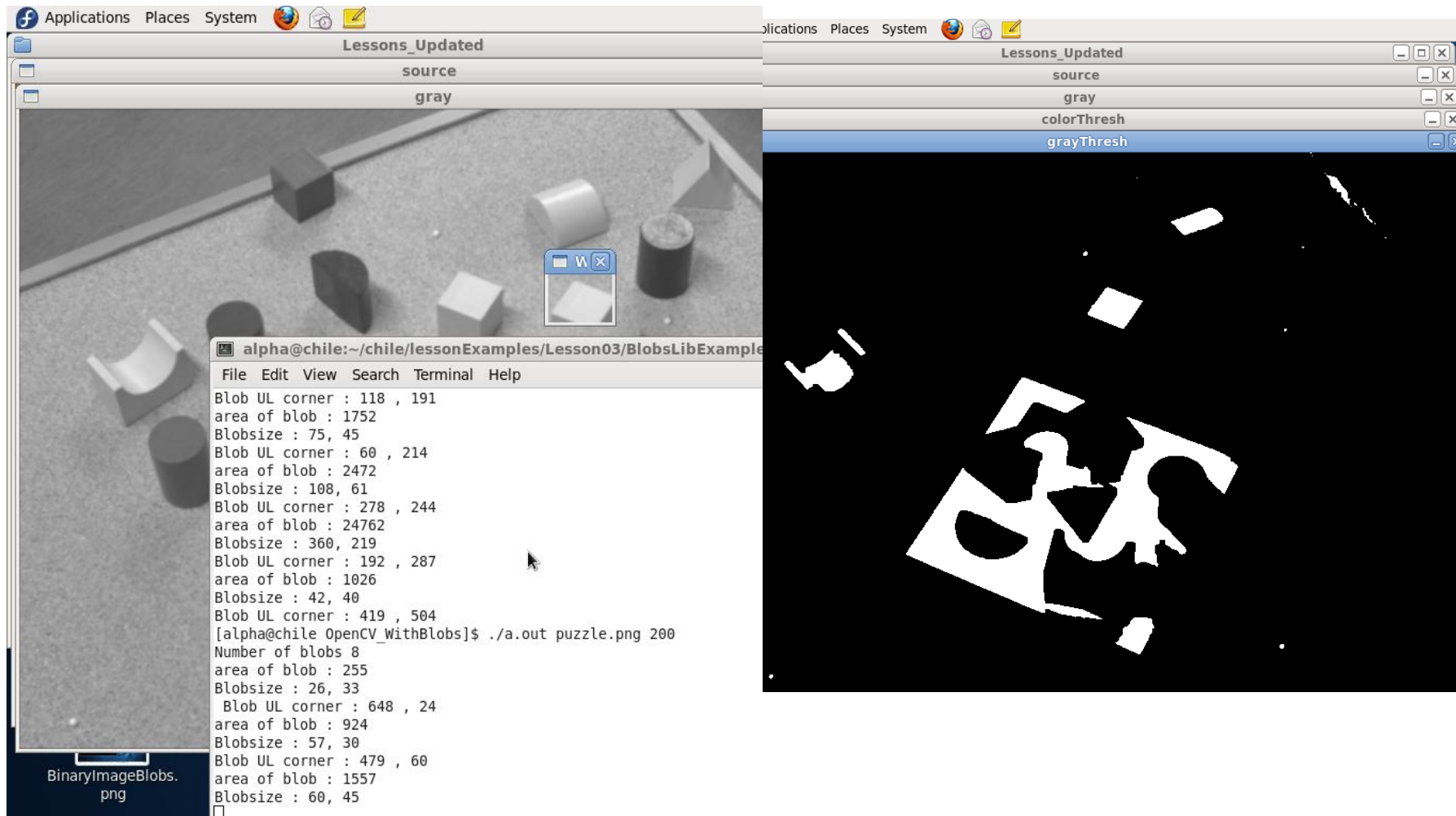
Another Sample Image

This simple algorithm can work with some "easy" images