Object-Shadow Pair Detection

By Group 4, Arjun Jauhari and Eric T. Jung

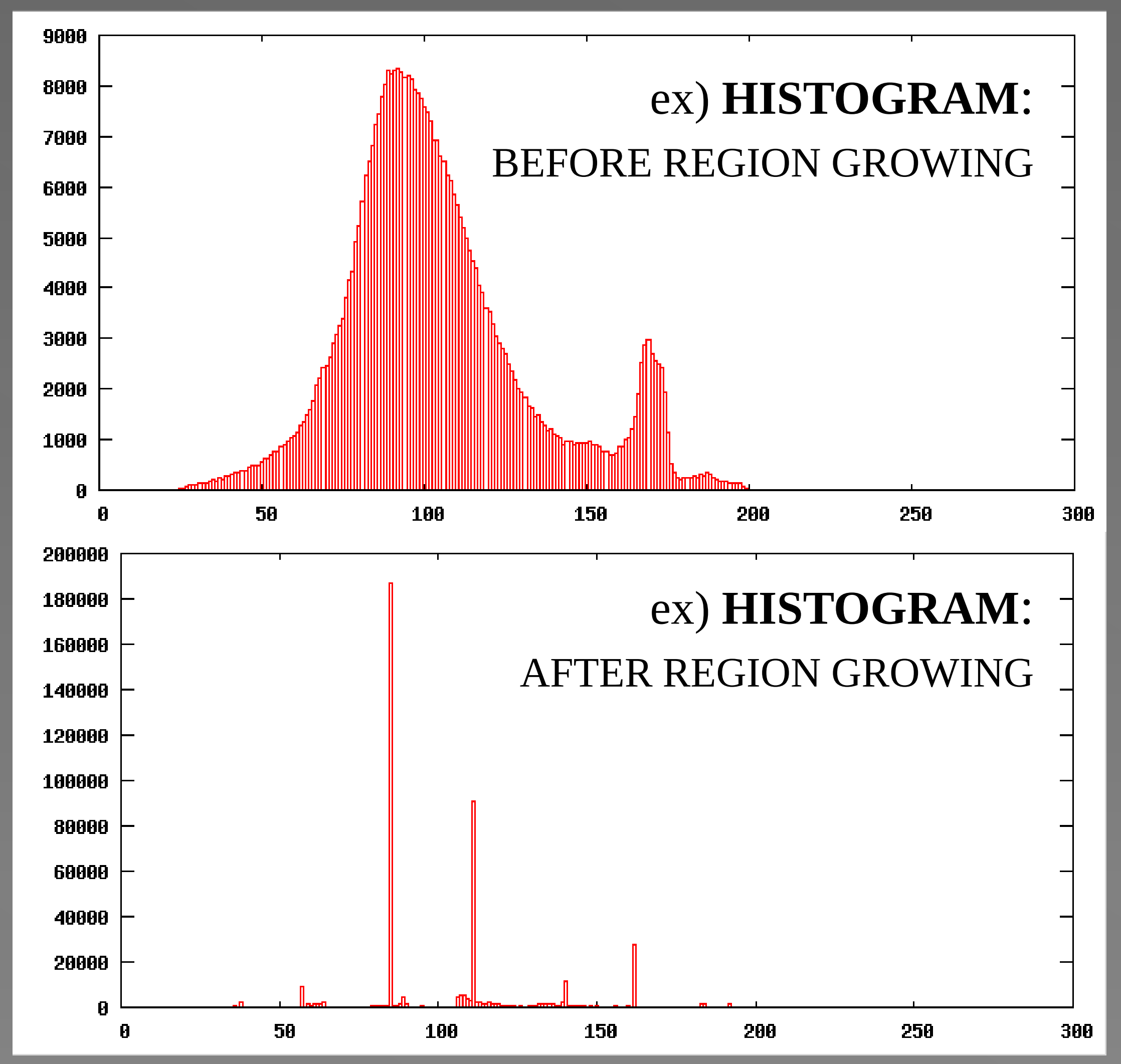
ABSTRACT

This project outlines a method of recognizing and segmenting an object-shadow pair, originally intended for calculating the angle of illumination. We attempt to tackle this segmentation task by composing different simple algorithms together. We designed an algorithm using region growing and adaptive multi-level thresholding to segment an object-shadow pair, and we use a visionX program vrdiff to evaluate and compute statistics on the accuracy of the algorithm. We gathered simplistic images with single image to test our algorithm first, and planned to test it further on more complicated images, but didn't have enough time. We were able to get workable segmentation algorithm, good enough, with adjustments, to be used for calculating angle of illumination for future work.

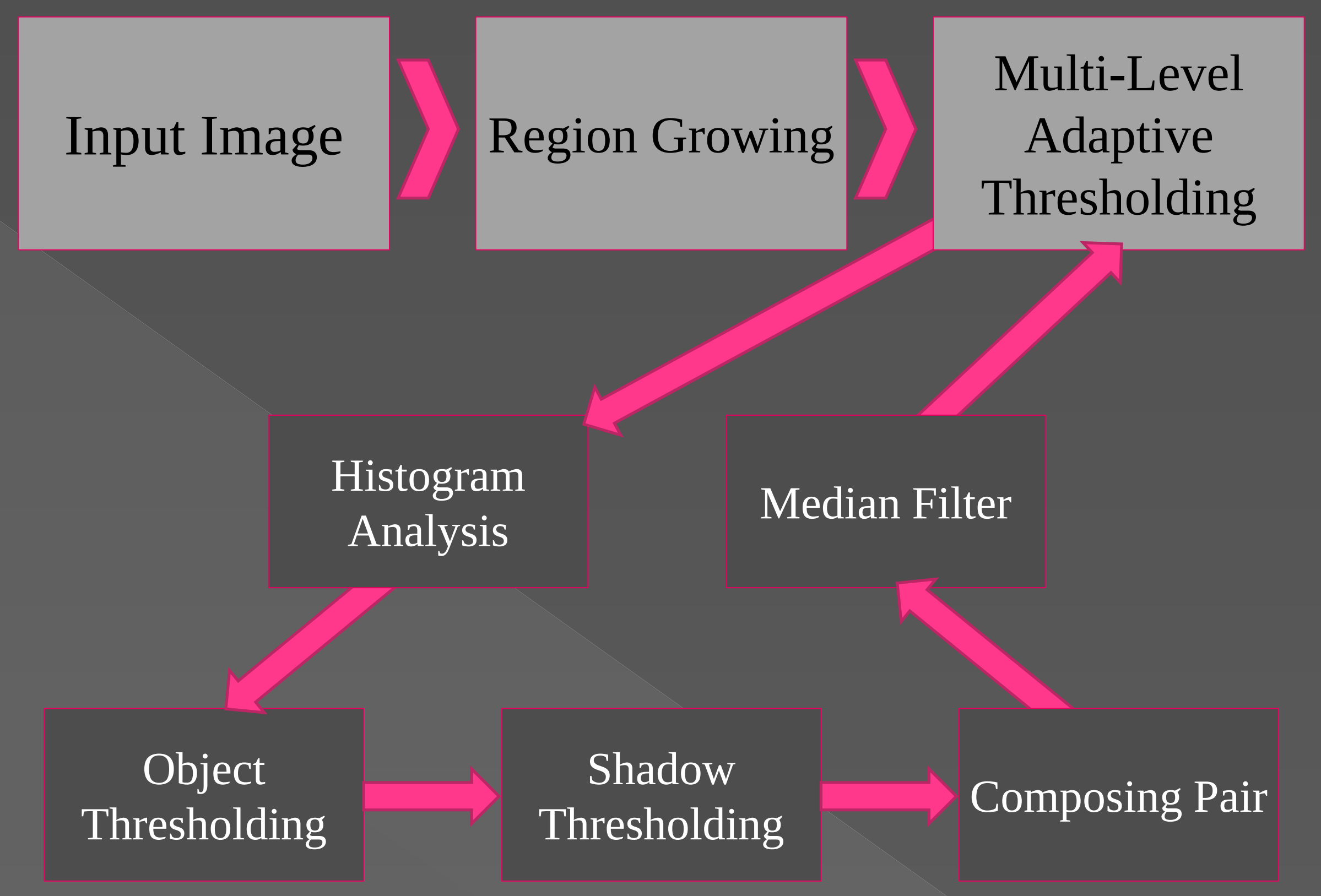
REGION GROWING

Very simple recursive region growing algorithm, adapted from code developed in lab 3

- Iterate over each pixel in image
- If the pixel is unlabeled, a recursive labeling function is called on the pixel location, with pixel value as parameter label; if it is labeled, move on.
- The recursive function labels the pixel at the location called with label, and looks around 4-neighbor pixels. For any neighbor that is not labeled and has pixel value within a hardcoded range of the label, recursive function is called again, but using the same label passed in originally, to prevent a moving target.



ALGORITHM OVERVIEW



MULTI-LEVEL ADAPTIVE THRESHOLDING

Histogram Analysis

- We search for a cutoff value on number of pixels to be classified as a peak. Any histogram bin whose value is higher than cutoff value is classified as peak.
- This search is done based on following constraints -
  - Number of peaks needs to be between bounded between maximum number of peaks and minimum number of peaks.
  - Distance between two peaks has to be greater than user specified constant(dist).

Object Thresholding

- After we have found peaks from Histogram Analysis step above, we use those peaks to find the peak belonging to object. We assume that the rightmost peak would be that of object since object tends to be most bright in the image. With this assumption we threshold the image with threshold set between rightmost two peaks.

Shadow Thresholding

- Similar to object thresholding, we assume shadow peak to be the leftmost peak. Thresholding is down with threshold set between the leftmost two peaks.

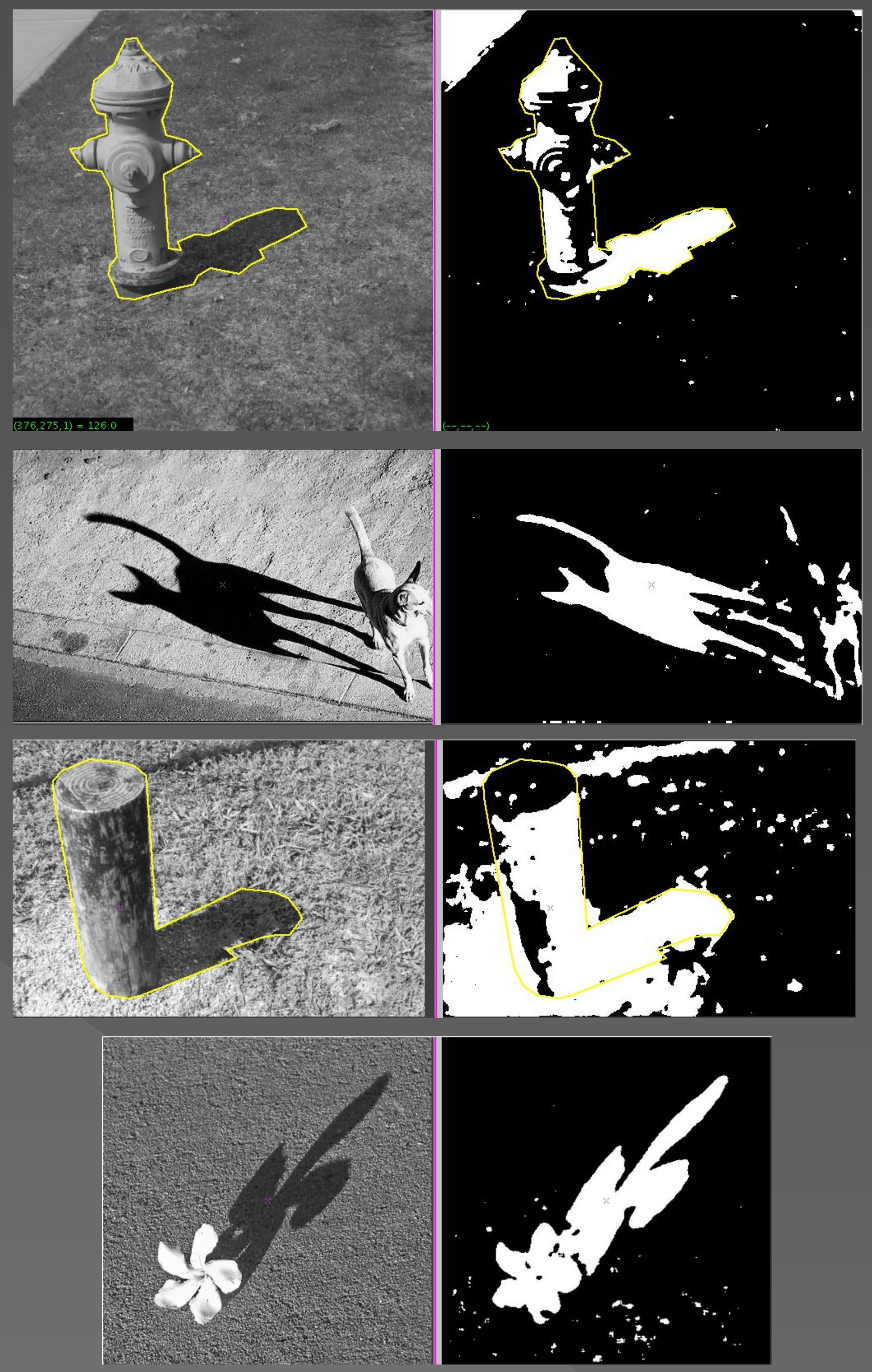
Composing Pair

- We simply OR each pixel location together, composing a doubly-thresholded image.

Median Filter

- We perform simple 3x3 median filtering on the OR'd image to filter out some noise.

RESULTS



EVALUATION METRICS

Given annotated ground truth A and segmented image B, we define following metrics, which are all calculated by vrdiff

Dice Similarity Coefficient

- 2 * | A n B | / (| A | + | B |)
- Indicates how "similar" two objects are. There is no ground truth for DSC.

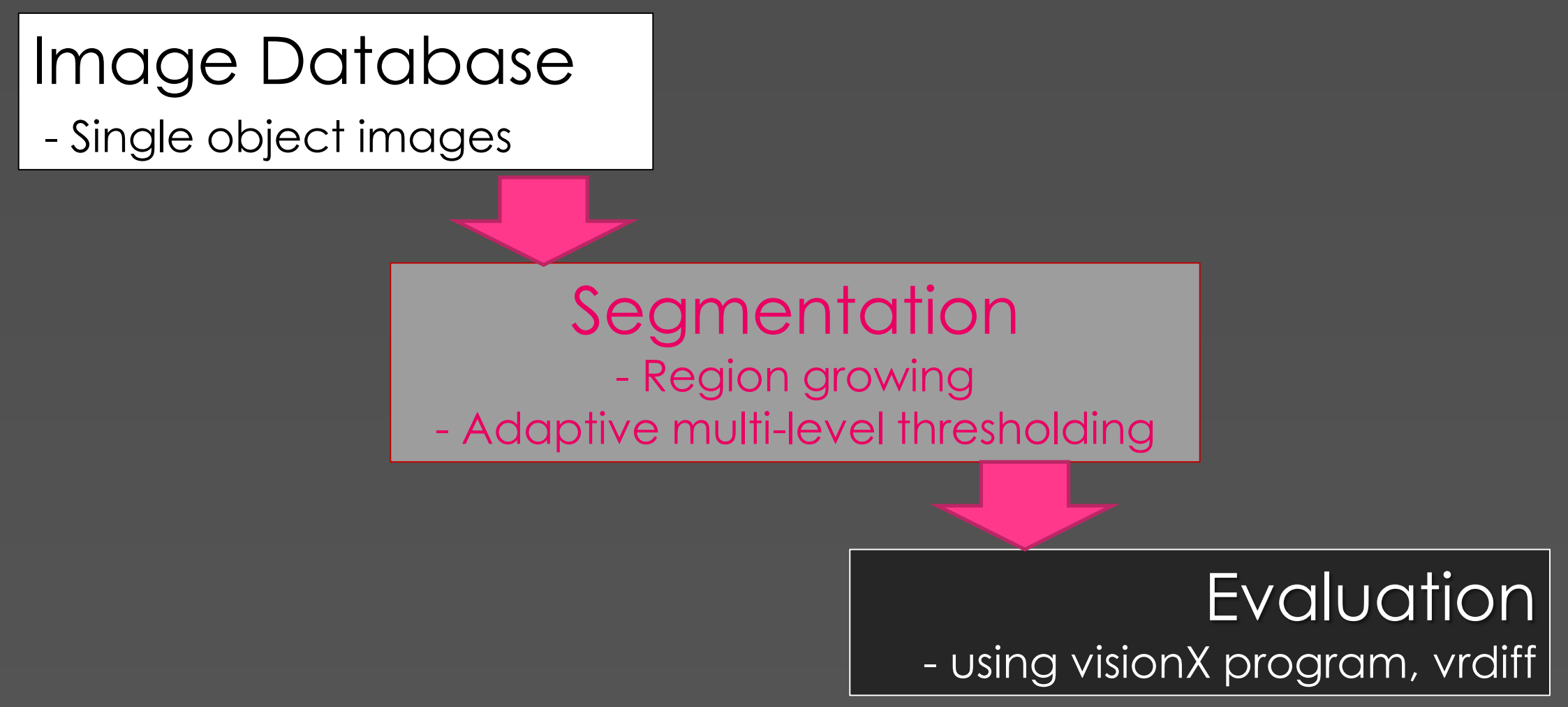
Sensitivity

- | A n B | / | A |
- Indicates % of object correctly segmented

Specificity

- ~ | A n B | / (~ | A |)
- Indicates % of background correctly segmented

PROGRAM FLOW



EVALUATION

	DSC	SENSITIVITY	SPECIFICITY
Hydrant	0.29	0.32	0.88
Dog	0.44	0.96	0.74
Wooden Post	0.37	0.39	0.62
Flower	0.85	0.98	0.96

CONCLUSIONS

Out of 4 examples we have listed, human visual system can probably distinguish 2-3. The metrics used for evaluation also tells a similar story, where the flower is segmented almost perfectly, with all percentages nearing100%.

In the case of dog, the majority of the error comes from specificity, where the background between shadow of the legs are annotated as part of object-region pair, where as the algorithm identified it (correctly, in some sense) as background rather than continuous part of the shadow.

In the case of hydrant, we have a huge section on the top left corner that has been identified as an object, due to how our thresholding works.

Wooden post was the hardest to segment, with low scores all around. Main reason for this was that the pixel difference between object and background was not significant enough.

As shown by each example chosen, the algorithm still has a lot of room for improvement. However, we were able to achieve this much using simple composition of two simple algorithms.

For future work, on the application side, there can be work done to use this segmentation algorithm for its intended use, to calculate the angle of illumination. On improving the algorithm side, perhaps one can perform closing with different parameters to completely fill the gaps between the edge regions.