Report Blatt 4, Advanced Part

Alexander Attinger, Yannic Kilcher

January 30, 2013

1 Watershed segmentation

We were asked to perform a watershed segmentation on a preprocessed image. We have done the preprocessing with the provided image of apples, as seen in Fig. 1

After the preprocessing, which consisted of various morphological operations, we could obtain a foreground and background label map. We combined these to give us a single label map to give to the watershed algorighm. In this map, as seen in Fig. 2, foreground pixels are labeled with intensity 255, background pixels are labeled with intensity 155 and unknown pixels are labeled with intensity 0.

The watershed algorithm iteratively fills the unknown pixels by expanding the already known regions. After the algorithm, we obtain the final label map, shown in Fig. 3.

As one can see, the algorithm selects more apples to the foreground, the smaller the initial disk radius, which is mainly due to more apples being labeled as foreground in the initial foreground map, which in turn results from the smaller disk size being able to "fit into" more parts of the apples. The watershed algorighm generally performs well, but a big problem is that once something is labeled as a certain region, the label will not change. This is problematic, when one cannot reasonably threshold it out in the initial label maps, as is the case with the middle part of an apple on the bottom left being labeled as background right from the start, which will never change.

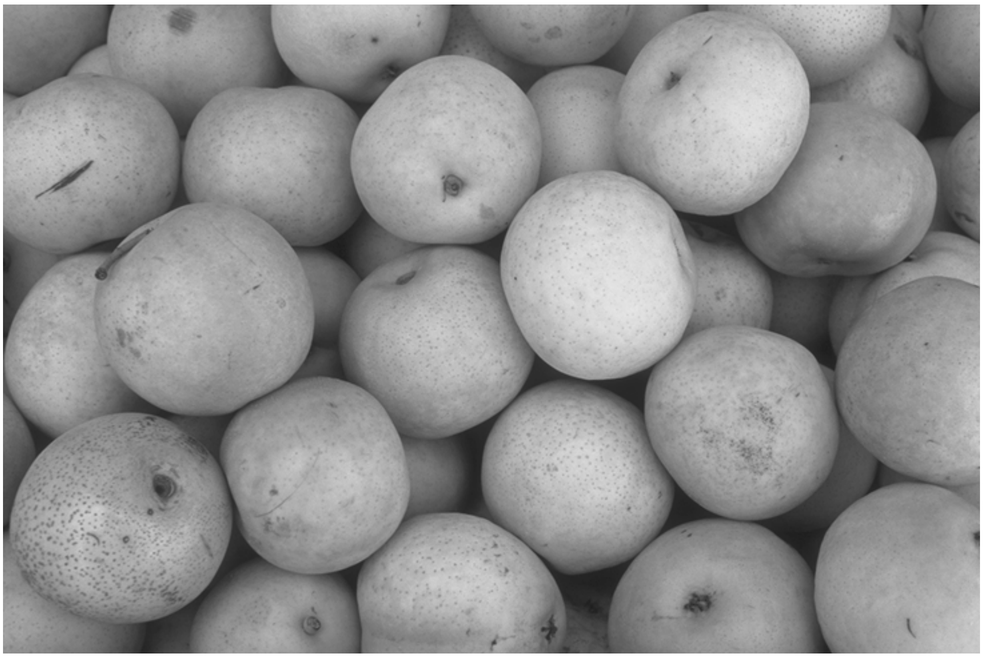
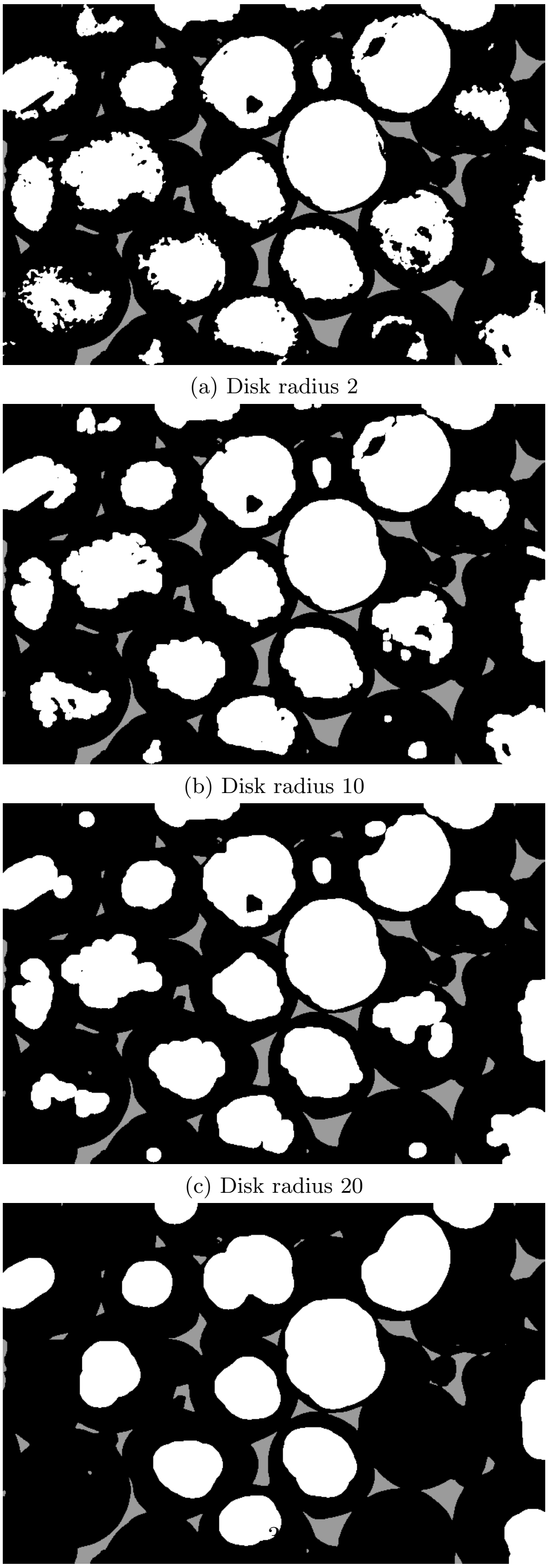


Figure 1: The original Image



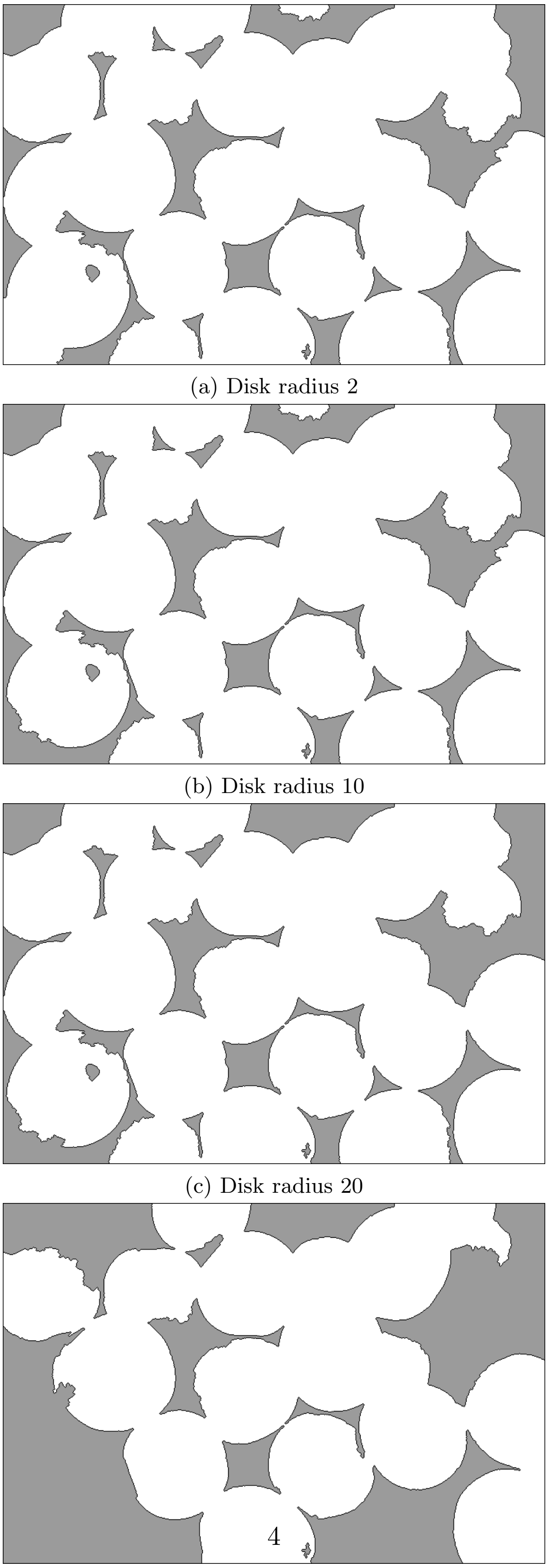
(a) Disk radius 2

(b) Disk radius 10

(c) Disk radius 20

(d) Disk radius 50

Figure 2: Label maps with various morphological operator sizes. Foreground thresholding at 160, background thresholding at 60



(a) Disk radius 2

(b) Disk radius 10

(c) Disk radius 20

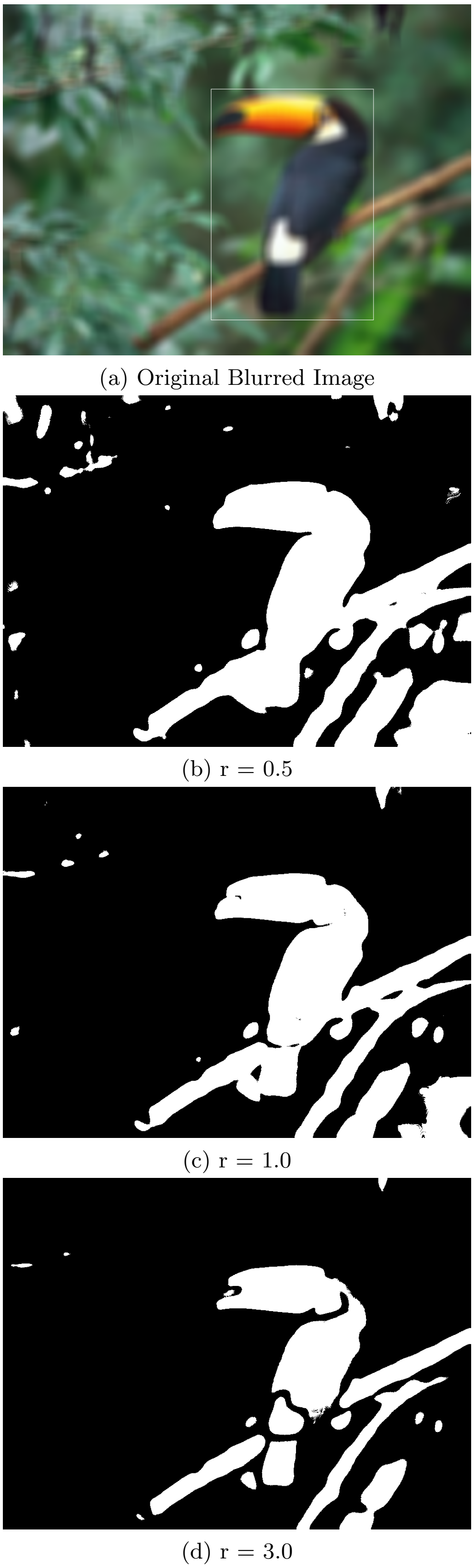
(d) Disk radius 50

Figure 3: Final label maps with various morphological operator sizes.

2 Maximum likelihood segmentation

We have implemented a gaussian maximum likelihood segmentation which learns from a bounding box marking foreground pixels. From that, we calculate two distributions - foreground and background - and then classify a pixel according to the ratio r of the two likelihoods. For quality reasons, we have first blurred the image with a size 25 gaussian kernel to reduce noise when training and classifying.

We have classified two images (Fig. 4/5) with different values of r. As one can see when comparing Fig. 5 to the grabcut segmentation of the same image in the report of the basic part, grabcut performs better than the ML approach, as grabcut also uses regional information about the pixels, whereas ML simply classifies according to color, not regarding the pixel's position in the image.



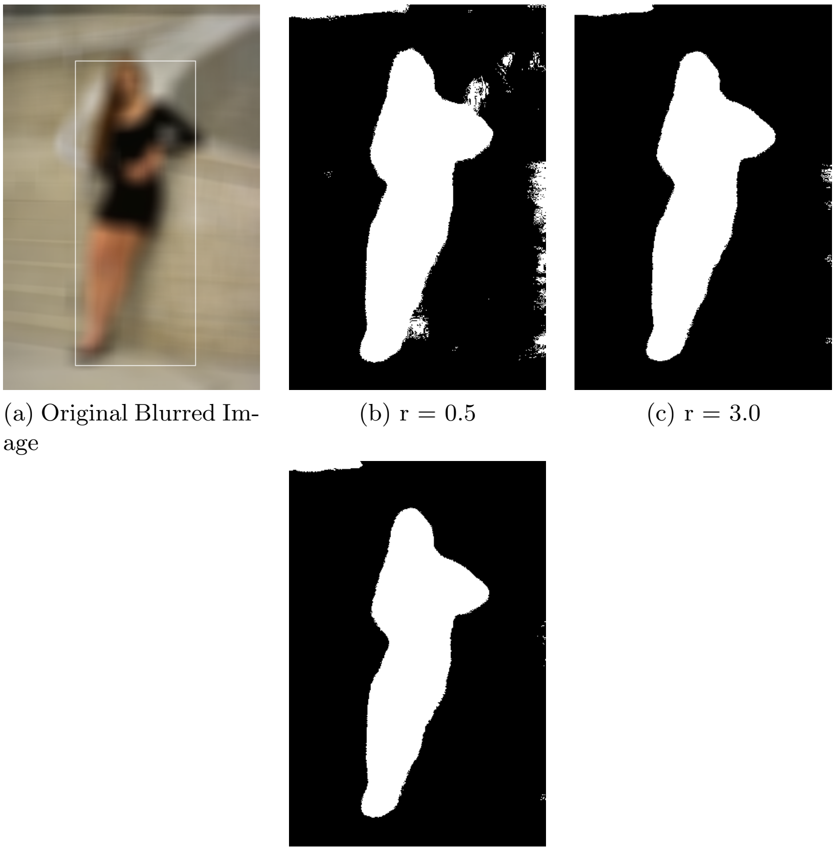
(a) Original Blurred Image

(b) r = 0.5

(c) r = 1.0

(d) r = 3.0

Figure 4: ML classification of the toucan image



(a) Original Blurred Image

(b) r = 0.5

(c) r = 3.0

(d) r = 5.0

Figure 5: ML classification of a person