**Critical Thinking 6**

Alec Meyer

Colorado State University Global Campus

CSC515

Dr. Jack Li

5/30/2025

**Image Segmentation**

Image segmentation is a crucial technique of image processing for computer vision applications. Image segmentation is generally done through thresholding or region-based segmentation. These two methods have their strengths and weaknesses depending on the use case for your application. Thresholding is one of the most common and straightforward methods of image segmentation, and it can be a powerful tool for separating foreground objects from the

background.

**Thresholding Segmentation**

Image segmentation thresholding is used to differentiate foreground images from their background despite the color intensities of foreground objects. Thresholding works under the assumption that the foreground and background have differing intensities, and a simple algorithm then converts every pixel in an image to black or white, depending on a specific global threshold. When using simple thresholding, the only hyperparameter a user has control over is the pixel intensity threshold at which a pixel would convert from black to white. For example, the black (left) water bottle presented in the attached script has a very low intensity, and the table that it is lying on also has a low intensity. If you were to use a global threshold value of 127 (half of 255), the entire water bottle would be converted to the same pixel as the background. However, if you were to tune the hyperparameter down to around an intensity of 35, the algorithm would see a difference between the water bottle and the table. However, an intensity of around 35 would fail to detect the white water bottle, as the algorithm would white out the white water bottle and the table.

**Adaptive Thresholding**

A technique for addressing the issues with a single threshold value is adaptive thresholding.

Adaptive thresholding considers a small neighborhood of pixels and determines the most optimal threshold for just that area (Rosebrock, 2021). This is a very powerful segmentation technique when an image has inconsistent lighting conditions or when foreground objects have varying intensities. For this project, adaptive thresholding is particularly important because the intensity of the background falls between the intensities of the two foreground objects. This means that no single global threshold will properly segment the image. Adaptive thresholding, however, can automatically set the threshold for pixels surrounding the white water bottle and choose a different threshold for pixels surrounding the black water bottle.

For the implementation of adaptive filtering in this project, I used two different adaptive filtering methods, mean and Gaussian. Mean adaptive thresholding sets the threshold of a neighborhood of pixels to the mean of that neighborhood. Gaussian adaptive thresholding sets the threshold to a weighted sum based on the Gaussian distribution (Yadav, 2024). When applying these algorithms to the image where the light source is facing the black water bottle, the results were pretty good. The outlines of the two water bottles are very defined and noticeable. However, when the light source is facing the white water bottle and the black one has more shadow on it, the algorithm still can outline the water bottles, but it is not nearly as defined. To improve the algorithm for the second image, I increase the neighborhood size. Increasing the size of the pixel neighborhood enabled the algorithm to view more surrounding pixels, thereby allowing it to handle shadows more efficiently. The shadows could be thought about as noise in this scenario, and increasing the neighborhood size diminished the impact of the noise.

**Conclusion**

Overall, thresholding, and specifically, adaptive thresholding, is a straightforward yet powerful tool for image segmentation. Adaptive thresholding leverages the strengths of global thresholding while overcoming its primary limitation: the need for a single global threshold value. Adaptive thresholding also provides greater flexibility to developers, as they can choose from various thresholding methods and neighborhood sizes, which can enhance performance in specific use cases.

**References**

Rosebrock, A. (2023, June 9). *Adaptive Thresholding with OpenCV ( cv2.adaptiveThreshold ) - PyImageSearch*. PyImageSearch. https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/

Yadav, A. (2025, April 18). OpenCV Adaptive Threshold - Amit Yadav - Medium. *Medium*. https://medium.com/@amit25173/opencv-adaptive-threshold-fae667b91984