Identification of Objects for Robotic Bin Picking Applications

Edward Venator

Department of Electrical Engineering and Computer Science

Case Western Reserve University, Cleveland, OH, Email: edward.venator@cwru.edu

# Abstract

This paper presents a method for segmenting an image of isolated and overlapping objects such that each segment contains only one object. Segmentation is accomplished by iterative morphological operations. The image is first thresholded to remove the background, then dilated to close holes in the objects. The dilated image is eroded with increasingly larger structuring elements to isolate individual objects. As objects are found, they are added to a labeled image and removed from the working image used for future erosions, simplifying successive searches.

Keywords

image processing, robot, eigenvector, principle component analysis, bin picking, morphology, object identification

# INTRODUCTION

Researchers have been studying the task of picking up shampoo bottle caps from a bin with a robotic arm. They use topographical data from a three-dimensional scanner and fit the data to CAD models of the bottle cap to determine the pose of the cap so that the robotic arm can pick it up.

In order to simplify the task of fitting CAD models to the 3D data, image processing must be used to segment the image into regions. Each region should contain one complete bottle cap. The bottle caps may be isolated from one another, or they may be touching or overlapping.

# Methods of Identifying Objects

In addition to the morphological technique described below, several other object identification techniques were explored for this project.

Correlation of template images with the image being searched can be an effective method of finding objects, provided the objects are not rotated or rescaled in the image being searched. However, given the three-dimensional nature of the objects in these images (bottle caps), correlation proved highly ineffective, even when many template images were tested. This procedure also becomes computationally intensive as more template images are introduced.

Some researchers [1] have been successful using eigenvectors and eigenvalues to create “eigentemplates” of the objects to be found. This technique is often used for facial recognition. Initial testing of this technique, using methods described in [1], proved ineffective for this application, and the calculations were very computationally intensive for the large number of template images being considered (26). One of the main drawbacks of this method is it is designed for matching an image that has already been registered (aligned and scaled) with a known set of templates. This task is to find an object in an image that may have been rotated, translated, skewed, or slightly resized. The only way to apply eigenvector techniques is with an exhaustive search of an area of interest or of the entire image, and both require a lot of computation. Eigenvector techniques proved completely unable to identify objects in this project, returning an unacceptably high number of false positives that resulted in a very low signal to noise ratio. The techniques may have been poorly implemented, but it is very likely that they are just ill-suited to this task.

# Preprocessing

Since the morphological algorithm described below classifies blobs by their size, input images must be prescaled so that the objects (bottle caps) are a standard size between images. The algorithm has some toleratnce for variation in object size within images and between images, but even with this algorithm optimally tuned, the objects must not vary in size by more than a factor of two, or groups of two or more objects could be mistaken for a single object. Prescaling was accomplished manually with photo editing software, but in an industrial environment with known camera parameters, it could be performed automatically, perhaps with the assistance of a known calibration pattern printed on the bin.

In addition, some images have uneven backgrounds that could be mistaken for objects. Since the task is to find objects in bins, the inside of the bin is the region of interest. As such, the image should be cropped or masked to limit the search to the inside of the bin. This was accomplished manually with photo editing software, but image processing techniques such as the Hough transform [3] could be used to find the edges of the bin and automatically define a region of interest.

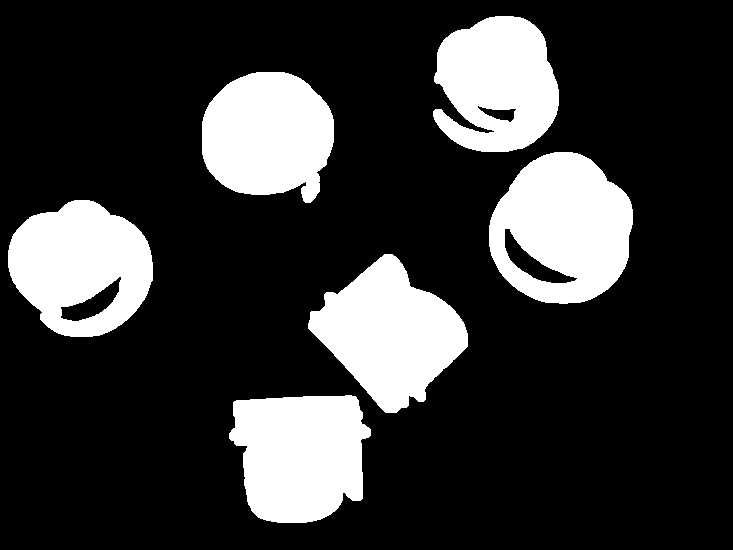
# Identifying Objects Using Morphological Methods

The technique that was ultimately used in this project uses morphological operations to separate objects from groups and label them. Once labeled, it is trivial to draw bounding boxes around the objects, find their centroids, or otherwise call out their positions to the next layer of processing.

If all of the objects in the image are isolated (none of them touch or overlap in the image), it is trivial to find the objects using thresholding and connectivity.

*Figure 1 shows an example of an image with isolated objects. It has already been rescaled to match the parameters of this algorithm.*

The first step of processing is to threshold the image, separating the objects from the background. Since the caps are white and the background is brown paper, it is possible to separate the caps from the background by thresholding the grayscale image. The best results were achieved when the grayscale image is contrast-adjusted using MATLAB’s imadjust() function and thresholded using Otsu’s method [3]. The bottle caps are retained and the background information is discarded, but there are still holes and discontinuities in the objects. In order to remove some of these, the image is dilated.

Figure 2 shows the result of thresholding Figure 1 using Otsu’s method and then dilating with a 6 pixel circular structuring element.

Now that the image has been prepared, the algorithm begins the search for objects. The result of the dilation is defined as the initial working image. The algorithm consists of a series of erosions by a circular structuring element. Each erosion’s structuring element is two pixels larger than the previous one. The erosions are not iterative, but instead performed on the same starting image, the working image. This is because a single erosion with a large structuring element is more effective at severing the links between touching objects than a series of erosions with small structuring elements. After each erosion, eight-connected regions are identified using MATLAB’s regionprops() function, which enumerates regions of white pixels in a binary image and extracts certain properties of those regions. Regions with an area smaller than a given threshold are known to be either single caps or erroneous blobs caused by noise in the image. If the region is larger than a minimum threshold, it is a cap.

If a region is determined to be a cap, it is dilated back to its original size (completing a morphological opening). The resulting blob is used as a mask to remove this cap from the working image. Removing found objects from the working image makes it easier to break blobs of multiple objects up into single objects and reduces computational operations as the algorithm progresses. The dilated blob is also added to a label matrix. The label matrix is a matrix of integers the same dimension as the original image. It stores labeled regions by filling in the pixel values with integers (labels) corresponding to each region. Since regions are discretized, it is easy to pick out individual regions for further processing or display. Finally, when an object is found, the size of the structural element is reset to the minimum size (two pixels). Because the working image is now different from the previous working image, there may now be isolated objects that were once part of blobs. Since morphological opening with a large structural element causes distortion and loss of detail, it is best to reset the size of the structural element and allow the algorithm to find any newly-isolated objects with minimal distortion.

The search terminates under one of two conditions. If at any time regionprops() cannot find any regions, it means that either all of the objects have been masked from the image or all of the objects have been eroded to nothing. In either case, it is impossible to find any more objects, and the search terminates. The other termination condition is a maximum size of the structural element, which in this case was 100 pixels.

Upon termination, the result is a label matrix identifying the areas of each of the found objects, and a list of the centroids of the found objects. Either can be used for further processing, such as creation of bounding boxes.

# Results and discussion

This method was tested on three images. The first image, shown in Figure 1 above, contained six isolated objects. The algorithm terminated after a single iteration, having found all six caps.

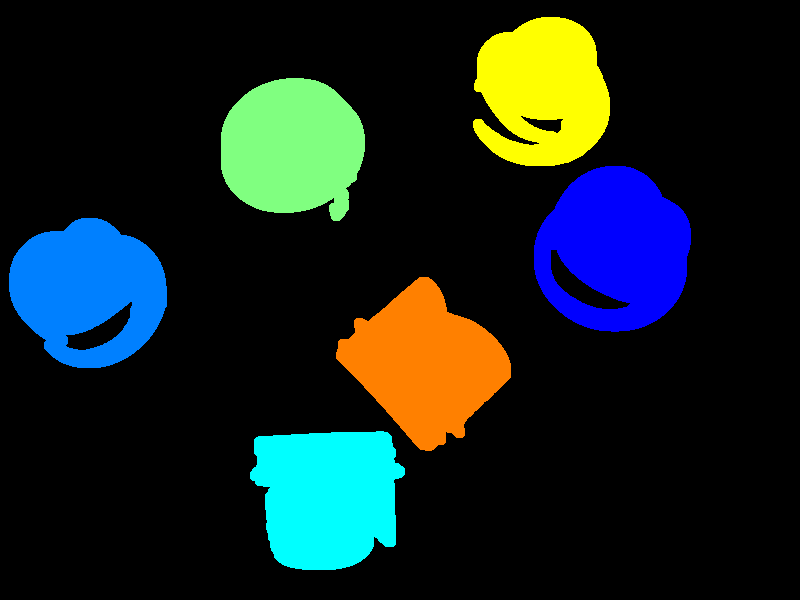


Figure 3 shows the results of running the search algorithm on Figure 1.



Figure 4 shows several partially-overlapping objects in a bin.

The second image contains more objects, and many of them are touching or partially overlapping. The algorithm was less successful with this input, but was still able to find sixteen objects in the image out of a total of twenty. Four of the “objects” that the algorithm finds are actually pairs of objects, and one object is found twice.

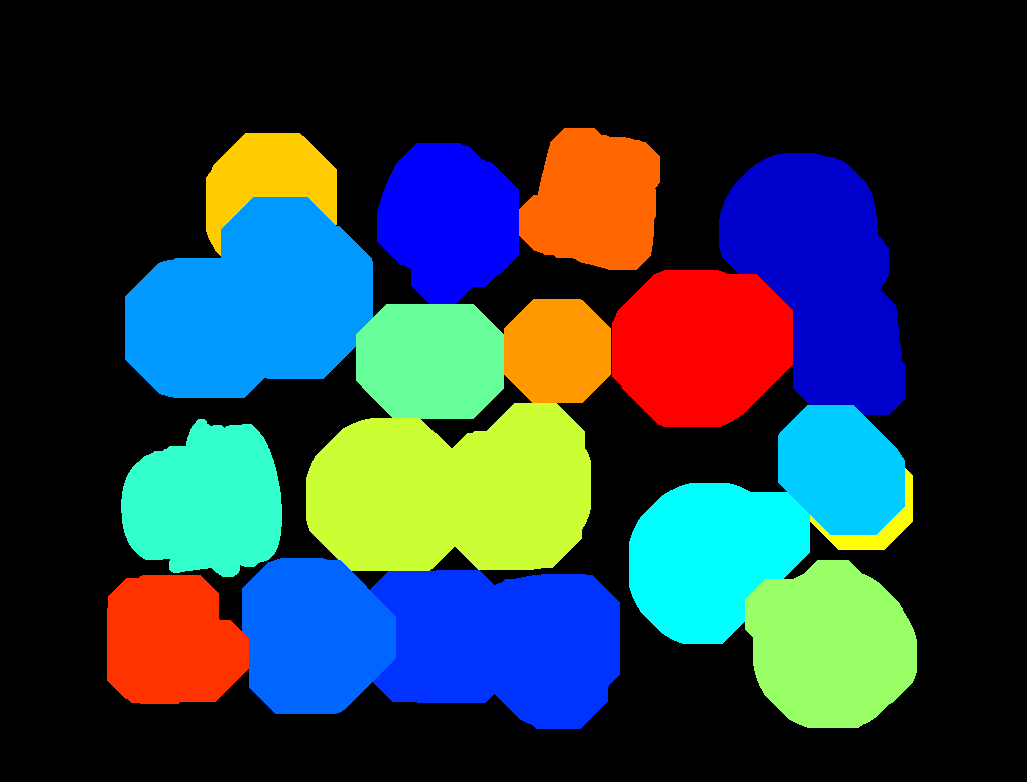
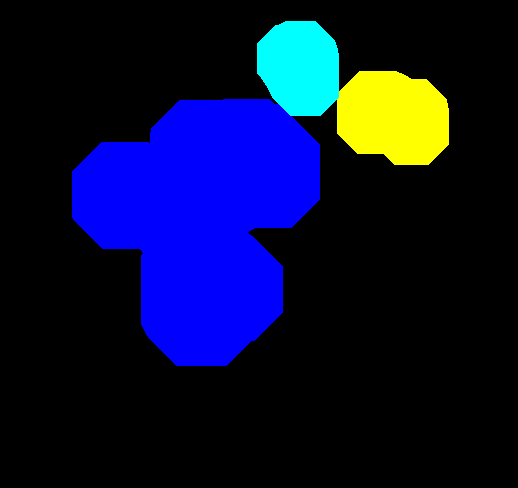
Figure 5 shows the results of running the search algorithm on Figure 4.



Figure 6 shows a pile of objects that overlap substantially.

The final image presents a much more difficult challenge. The caps are overlapping much more than in the previous images, and as a result, the algorithm has considerable difficulty identifying the objects. As shown in Figure 7 below, only two objects are correctly identified. A third blob contains five objects.

Figure 7 shows the results of running the search algorithm on Figure 6.

# Applying This Method

It is evident from the results above that this algorithm is suitable for isolated bottle caps. The results from the second image (partially overlapping caps in a box) would probably also be acceptable. However, the results for the third image show that this algorithm does not work for large piles of caps.

To apply this algorithm in an industrial situation, several accommodations must be made. First of all, it is necessary that the background be a color that contrasts with the objects being searched for. In these example images, the brown paper background contrasted with the white caps.

In addition, the objects must be in a single layer. If the objects are on a table or conveyor belt, this could be accomplished by vibrating the table to separate any piles. Unfortunately, it would be difficult to separate piles of objects in a box.

Summary

A morphological method of segmenting objects in an image was presented. This method was applied to three pictures of bottle caps, and the results were demonstrated. The method was completely successful at segmenting isolated objects, and somewhat successful at segmenting objects that partially overlap. The method was unsuccessful at segmenting objects that overlapped substantially or were in piles. These results were discussed and applications of the method were suggested.

Future work on this method would include experimenting with different shaped structural elements. Using different structural elements might separate the objects without eroding as much of the detail of the objects. This would allow more objects to be identified, and would also create make more accurate masks.

Other future work would be related to tailoring the output to match the needs of the 3D CAD model matching system. The labeled regions may need to be dilated slightly to cover the entire cap in the image. Other transforms may be needed to fill in any hole or gaps in the regions. One trivial solution would be to dilate the regions slightly and then create a bounding box around the regions.

Acknowledgments

This work was performed in the context of Professor Frank Merat's EECS 490 Image Processing class.

Chris Roberts provided valuable insight into object recognition techniques.

References

|  |  |
| --- | --- |
| [1] | O. Ghita and P. F. Whelan, "Object recognition using eigenvectors," *Proceedings of SPIE,* pp. 85-91, 1997. |
| [2] | S. Serrano, "Eigenface Tutorial," Drexel University, 23 August 2003. [Online]. Available: http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm. [Accessed 16 December 2011]. |
| [3] | R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM,* vol. 15, no. 1, pp. 11-15, 1972. |
| [4] | N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 9, no. 1, pp. 62-66, 1979. |