

JBoost Optimization of Color Detectors for Autonomous Underwater Vehicle Navigation

Christopher Barngrover, Serge Belongie, and Ryan Kastner

University of California San Diego,
Department of Computer Science

Abstract. In the world of autonomous underwater vehicles (AUV) the prominent form of sensing is sonar due to cloudy water conditions and dispersion of light. Although underwater conditions are highly suitable for sonar, this does not mean that optical sensors should be completely ignored. There are situations where visibility is high, such as in calm waters, and where light dispersion is not significant, such as in shallow water or near the surface. In addition, even when visibility is low, once a certain proximity to an object exists, visibility can increase. The focus of this paper is this gap in capability for AUVs, with an emphasis on computer-aided detection through classifier optimization via machine learning. This paper describes the development of color-based classification algorithm and its application as a cost-sensitive alternative for navigation on the small Stingray AUV.

Keywords: Stingray ; AUV ; object detection ; color ; boosting

1 Introduction

The goal of this paper is to use the Stingray platform to investigate object detection and classification as a basis for navigation. Reliable navigation on small AUVs is challenging in the absence of large and expensive sensors for estimating position. Using vision to detect and classify objects in the environment can be a source for estimating relative position. The target object can be used as a destination or could act as a path for the vehicle to follow [1]. The focus of this research is on developing robust object classifiers for specific targets based on color. The movement of the water and changes in lighting due to refraction and light dispersion cause colors to blur and change. In order to overcome these difficulties, we use a boosting algorithm to optimize the color classifier and improve the detector capability.

The target destination objects are three different colored buoys, anchored with relatively close proximity and varying depth. The buoy colors, chosen for their contrast with an underwater environment, are orange, yellow, and green in decreasing order of contrast. The green buoy should be more difficult to detect since it is most similar in color to the background. Once the algorithm can correctly detect and classify the target buoy, the vehicle demonstrates the

navigation capability by approaching and touching the buoy. The path or bearing objects are orange pipes, which are anchored to the bottom. In some cases there are two pipes with different orientations in the same location. The vision algorithms detect and classify the pipe and then estimate the orientation. The vehicle demonstrates the vision-based navigation capability by centering over the pipe and altering its heading based on the estimated orientation. When there are multiple pipes, the vehicle must decide which direction to navigate. The two target types are shown in Figure 1 below.

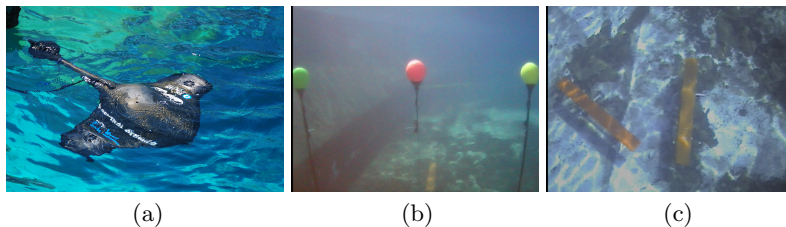


Fig. 1. (a) Stingray AUV. (b) Destination buoy objects. (c) Bearing pipe objects.

It turns out that the boosting of the classifiers for the buoys and pipes greatly improves the detectors. For the pipe, we show that the bearing estimation becomes extremely accurate as well. We implement the optimized detectors and bearing estimator on the Stingray, which is able to navigate to the correct buoy and change bearing based on the pipe with high reliability.

The remainder of this paper is organized as follows. In Section 2 we discuss related work, while in Section 3 we describe our process for developing a classification algorithm. In Sections 4 and 5 we focus on the specific targets of the buoy and pipe, providing results from the final algorithms for each. Finally, in Section 6 we conclude by discussing the aspects of this research that are novel and the promising directions for future work.

2 Related Work

There has been an increase of research in vision-based navigation for underwater vehicles in recent years. Most of the research focuses on avenues that do not parallel the work in this paper, but there are some similar efforts.

The papers that use landmarks as reference points for underwater navigation are most similar. The work of Yu *et al.* [7] uses yellow markers and colored cables for AUV navigation by thresholding the UV components of the YUV color space, which is similar to the baseline methods for this paper. Another method thresholds on the RG components of the RGB color space to detect yellow sensor nodes, as presented by Dunbabin *et al.* [3]. In the research by Soriano *et al.* [6]

an average histogram is created for each target, which is compared to a region of interest for classification.

Cable or pipe tracking is another task, which is heavily researched in terms of vision-based systems. The work of Balasuriya *et al.* [1] shows a method of using Laplacian of Gaussian (LoG) filters to detect the edges of the pipe. Foresti *et al.* [4] use a trained neural network to recognize the pipeline borders, while Zingaretti and Zanoli [8] use vertical edge detection in horizontal strips and contour density within the strips to detect the pipe.

These papers avoid much of the underwater difficulties, which cause colors to change based on light absorption, by attaining proximity to the target. We show that without boosting, a simple color classifier is not sufficient on our test data set, which includes images of the targets at substantial distances and under varying lighting conditions.

3 Developing a Classification Algorithm

The process of developing the classification algorithm generally starts with choosing a feature set to describe the target. The feature chosen for these targets is color. The Hue-Saturation-Value (HSV) color model is used for its separation of brightness from the hue and saturation pair. Because of this isolation of the brightness element of a color, a single object is more reliably detectable under different lighting conditions. The more common Red-Green-Blue (RGB) color model is an additive model, which makes it difficult to identify the same color under different lighting conditions [2].

The boosting algorithm requires a large number of examples in order to optimize the decision tree. For the HSV color classifier, we labeled individual pixels as positive or negative in terms of the target. The examples, which number in the hundreds of thousands, are then inputs into the boosting algorithm.

For this research, the LogitBoost form of boosting is used via the JBoost software package. The JBoost application expects the input examples in a standard format with classifier data and a label. JBoost can output the resulting decision tree visually as well as in Java or C code.

4 Buoy Detection

The buoy targets have the same size and shape, only differing by color. To develop the algorithm, we focus first on the orange buoy. Once an algorithm is developed, including the pixel level optimized decision tree and post processing, we can train the classifier for the other buoys. The final algorithm will have a pixel decision tree for each color to create a binary image. The binary image will be post processed in the same way for each color. The goal is to accurately estimate the location of the designated buoy in the image and use the distance from the buoy to the center of the image as a heading offset for the Stingray vehicle.

4.1 Baseline

There must be a baseline algorithm in order to determine the improvements provided by using boosting to optimize the decision tree for the HSV classifier. The baseline in this research is a simple HSV thresholding, which was previously implemented on the Stingray. An HSV estimation of the color orange in the buoy is extended to provide a range for each of hue, saturation and value, which was tuned over many iterations to achieve the best possible threshold range. The range is used to determine if a pixel is positive or negative, thus creating a binary image, which is used without post processing to estimate the center of the buoy based on the centroid of the positive pixels.

The metrics used to compare algorithms are the true positive rate (TPR) and false positive rate (FPR). There are two sets of images from two different environments. The first environment is a large anechoic pool, which is 300 ft by 200 ft by 38 ft deep. The other is a small above ground pool, which is 10 ft in diameter and 4 ft deep. Both pools are situated outside in natural lighting. For each environment there is a set of images for training the classifier and a set of images for testing the resulting classifier. Both image sets have examples of the buoy from different distances as well as images with no buoy present. To determine TPR and FPR, we label the center of the buoy in each test image, as well as the edge of the buoy. The distance between these points provides a threshold for the correctness of a center estimation. The baseline TPR is 0.45 and 0.18 for the Tank and the Pool respectively, while the FPR is 0.55 and 0.45.

4.2 Post Processing

Since the boosted classification algorithm is for individual pixels, the output is a binary image without clearly defined object boundaries and with extraneous positive or negative pixel noise. The goal of the post processing techniques used in this research is to prepare the binary image for the best possible estimation of the location of the buoy.

We start by using one iteration of opening, which is erosion followed by dilation, to remove noise in the binary image. Next we use two iterations of closing, which is two dilations followed by two erosions, to fill binary objects containing gaps. Then the smoothing algorithm via Median blur with a 7x7 kernel creates smooth edges of binary objects in the image. Finally, we use the convex hull algorithm to approximate the shape of the binary object with only convex corners, which provides more complete binary objects in situations where part of the target is not correctly classified.

4.3 Boosting HSV

As described in Section 3, the first step to boosting the HSV classifier is labeling examples. The pixel examples are given as input to JBoost, which outputs a complex decision tree in a C code function. The function provides a score for a given pixel, which is labeled as a one or zero based on a threshold.

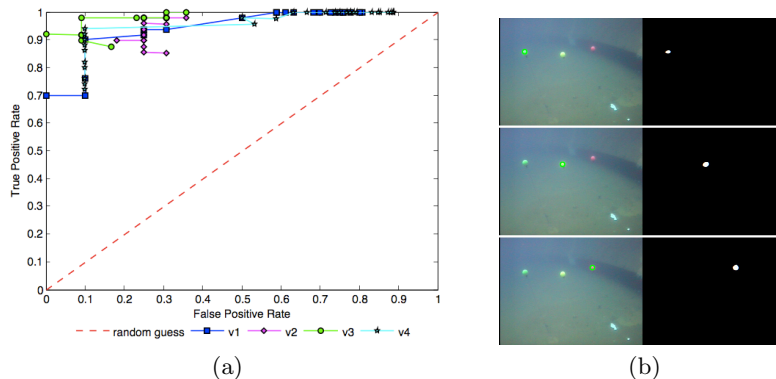


Fig. 2. (a) The ROC curves for four versions of the buoy classifier on the test image set from the tank environment. (b) Example of classifying specifically for different color buoys independently. The green circles show the estimated centers for each buoy.

In order to determine the threshold that provides the best output, we look at the receiver operating characteristic (ROC) curve for thresholds from -2.0 to 5.0 over 0.1 increments. Since the threshold determines the status of a pixel and the performance of the classifier is determined by the accuracy of the center estimation, the generated ROC curve is not a smooth curve. The tank is large and representative of an ocean environment in terms of acoustics and reflectivity, while the pool is small with reflective walls and bottom. The two environments are distinct enough that when we label extra examples for the pool, we ultimately overfit causing reduced performance for tank images. The simple solution is to develop target classifiers for the environments independently.

We start with the tank environment by generating a decision tree, which we use on our test image set to produce the ROC curve and choose the best threshold value. Based on the results at this threshold, additional labeling may improve the classifier. Figure 2 shows the ROC curves from four such iterations of the decision tree. The best results are at the threshold of 3.6, which gives a TPR of 0.98 and a FPR of 0.18, and the threshold of 4.2, which gives a TPR of 0.92 and a FPR of 0.0.

We follow the same iterative sequence for the pool environment, which is much more challenging because of its small size and shallow depth. The two best thresholds are 0.7, which gives a TPR of 0.68 and a FPR of 0.26, and the threshold 1.7, which gives a TPR of 0.61 and FPR of 0.05. These results are not as reliable as the tank results, but they are still a substantial improvement over the baseline.

4.4 Results

The same technique described in Section 4.3 can be applied to the other two buoy colors to create decision trees for classifying the pixels. The post processing

techniques are the same for each color buoy. This means that the algorithm will switch between the decision trees based on the target buoy. Figure 2 shows the processing of the same image while looking for each of the different color buoys.

When combining the results of the three buoy classification algorithms on the test image set, we can calculate the total TPR and FPR for the overall algorithm as 0.84 and 0.16 respectively. The relatively low quality of the classifier for the green buoy reduces the overall result.

In practice the Stingray is able to reliably detect the designated target buoy at approximately six frames per second and the detection becomes more reliable as the Stingray approaches the buoy.

5 Pipe Detection

The pipe is an interesting target because it provides a bearing for navigation. There can be two pipes leading to different destinations, as shown in Figure 1, which means the algorithm needs to be able to classify multiple pipes in a single image. After determining that a binary object is a pipe, the algorithm must calculate the orientation. The goal is to use the orientation of the pipe as a target heading for the Stingray vehicle.

5.1 Baseline

The baseline for the pipe, similar to the buoy, is a simple HSV threshold used to create a binary image on which a custom algorithm, using least squares estimation, attempts to determine the orientation. This orientation estimation technique is not dependable and is only used in the baseline algorithm.

The same metrics are used for the pipe results as are used for the buoys. The main difference is that there are no examples from a secondary environment. This makes the classification problem slightly easier, so that the problem of estimating orientation can take focus. The baseline for the Tank is a TPR of 0.74 and a FPR of 0.16.

5.2 Classification

The pipe, like the buoy, has a unique color which makes for a useful classifier. The same process of labeling images and inputting the examples into JBoost to optimize a decision tree ultimately outputs a function for scoring individual pixels of the image. The same post processing techniques from Section 4.2 are applied to the pipe binary images to create smooth and closed binary objects.

The version of the decision tree that produces the best ROC results has two thresholds with a trade off between TPR and FPR. Both of these thresholds provide very reliable rates, -0.3 give a TPR of 0.97 and a FPR of 0.02, while the threshold 0.7 gives a TPR of 0.95 and a FPR of 0.01.

5.3 Bearing Estimation

The overall goal of the pipe detection is to determine the orientation of the pipe to be used as a bearing for navigation purposes. Therefore, with a binary object found, only the edges of the object are actually pertinent. The Canny edge detector, with threshold values of 50 and 150 pixels, is applied to the binary image and the output contains only the edges of all binary objects.

With only edges remaining, the Hough Transform can be used to easily estimate the straight lines in the image. We use the Probabilistic Hough Transform (PHT) due to its ability to combine similar lines with a gap between them [5]. We use a ρ of one pixel and a θ of $\frac{\pi}{120}$ or 1.5 degrees. Our threshold is set at 30 pixels, with an acceptable line segment length of 20 pixels and an acceptable gap of 20 pixels.

Often times the output from the PHT has extraneous line segments. The goal of the pruning portion of the algorithm is to reduce all the line segments from the Hough Transform down to the two per pipe that represent the long edges of the pipe. This is broken into two steps, starting with merging all line segments that are close to collinear. The next step is using the property of parallelism to remove extraneous line segments. Figure 3 shows three scenarios where different tests of parallelism remove extraneous line segments.

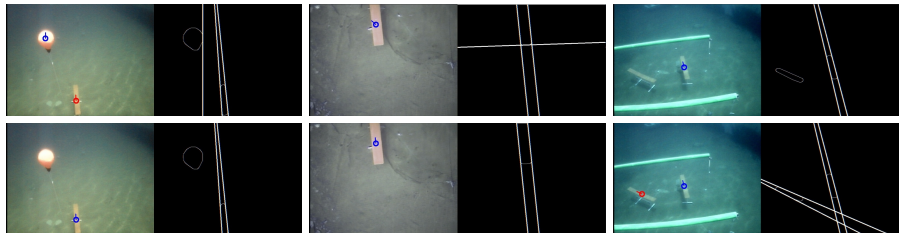


Fig. 3. Examples of the three algorithms of the pruning stage. The blue and red circles with lines show the estimated centers and orientations of the pipes.

5.4 Results

The important result of the pipe detection is the ability to estimate the orientation of the pipe with great precision, in order to provide the vehicle with useful bearing. Of course, detecting the location of the pipe is necessary to allow for the bearing estimation, which we have shown to be very reliable.

In order to quantify the accuracy of the bearing estimation, the edges of the pipes are labeled in the test image set and then compared to the algorithm's estimate. The average error with standard deviation for the baseline algorithm is $9.0^\circ \pm 14.6^\circ$ compared to $0.7^\circ \pm 0.8^\circ$ for the hough transform based algorithm.

In practice the Stingray is able to process the images at five frames per second, allowing the vehicle to center itself over the pipe and estimate the orientation. The vehicle then rotates to match its heading with the orientation of the pipe, and navigates in that direction.

6 Conclusion

This paper presents a method for using object detection and classification of target objects to aid in navigation for AUVs. The color classifier is one unique element of this research, as it is not common in underwater applications. Also, the use of boosting algorithms to optimize the classifier greatly improves on previous work. We incorporated the use of post processing techniques to make identifying the center of the target objects more reliable. We also showed a technique for calculating the orientation of up to two pipes simultaneously, and with high precision.

The result is two classification algorithms that are more efficient than the baseline algorithms of simple thresholding. We demonstrated these algorithms on the Stingray AUV, which navigates towards and touches a specific color of buoy and changes heading based on the pipe.

The process we presented for creating an optimized classifier via boosting can be applied to other targets and with classifiers other than color. The complex and dynamic properties of underwater environments cause these classifiers to be very specialized, which naturally leads this research towards efforts in adaptive learning to improve a classifier in real time for changing environments.

References

1. Balasuriya, B.A.A.P., Takai, M., Lam, W.C., Ura, T., Kuroda, Y.: Vision based autonomous underwater vehicle navigation: underwater cable tracking. In: OCEANS Proceedings, pp. 1418–1424 (1997)
2. Cheng, H. D., Jiang, X. H., Sun, Y., Wang, J. L.: Color image segmentation: Advances and prospects. *Pattern Recognition* 34, 2259–2281 (2001)
3. Dunbabin, M., Corke, P., Vasilescu, I., Rus, D.: Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In: IEEE Int. Conf. on Robotics and Automation, pp. 2091–2098 (2006)
4. Foresti, G.L., Gentili, S., Zampato, M.: A vision-based system for autonomous underwater vehicle navigation. In: OCEANS Proceedings, pp. 195–199 (1998)
5. Kiryati, N., Eldar, Y., Bruckstein, A. M.: A probabilistic Hough transform. *Pattern Recognition* 24, 303–316 (1991)
6. Soriano, M. and Marcos, S. and Saloma, C. and Quibilan, M. and Alino, P.: Image classification of coral reef components from underwater color video. In: OCEANS Proceedings, pp. 1008–1013 (2001)
7. Yu, S.C., Ura, T., Fujii, T., Kondo, H.: Navigation of autonomous underwater vehicles based on artificial underwater landmarks. In: OCEANS Proceedings, pp. 409–416 (2001)
8. Zingaretti, P., Zanolli, S.M.: Robust real-time detection of an underwater pipeline. *Engineering Applications of Artificial Intelligence* 11, 257–268 (1998)