On the Performance of Color Tracking Algorithms for Underwater Robots under Varying Lighting and Visibility

Junaed Sattar and Gregory Dudek
Centre for Intelligent Machines
McGill University
Montréal, QC H3A 2A7, Canada.
Email: {junaed,dudek}@cim.mcgill.ca

Abstract— **We consider the use of visual target tracking for autonomous steering of an underwater robot. In this context, we consider a performance comparison for three key visual tracking algorithms used for servo control. We present a comparative study of the performance in underwater environments of three tracking algorithms that are widely used in vision applications. Variations in illumination, suspended particles and a resulting reduction in visibility hinders vision systems from performing satisfactorily in marine environments; at least not as well as they do in terrestrial (*i.e.* non-underwater) surroundings. Our work focuses on quantitatively measuring the performance of three color-based tracking algorithms– color blob tracker, color histogram tracker and mean-shift tracker, in tracking objects underwater in different levels lighting and visibility. We also present results demonstrating the effect of suspended particles underwater, and in conclusion we summarize the three tracking algorithms by comparing their pros and cons.**

I. Introduction

We consider the use of visual target tracking for autonomous steering of an underwater robot. In this context, we consider a performance comparison for three key visual tracking algorithms used for servo control. Using vision sensors in underwater applications has certain desirable properties, compared to other more 'traditional' sensing methods like sonar or infrared (IR). Vision is a passive sensing medium, unlike sonar or IR sensors, which makes it energy-efficient (in spite of the power required for the sensor, *i.e.*camera). This purely passive operation of vision also makes it desirable in marine biology applications such as monitoring coral reefs and registering fish population in particular habitats in the ocean. Along with the need for a sensor to collect data, autonomous underwater vehicles are becoming an increasingly effective and robust technology [1][2]. The majority of these vehicles have no working vision sensors for achieving autonomous operation, mostly relying on active and passive sonar to navigate. In recent work, underwater vehicles have been deployed for shallow-water navigation using vision as the primary sensing mechanism [3]. One of the main causes of vision being neglected, is the effect underwater objects and the water medium itself exerts on light beams. We take a closer look at these effects, particularly the effects on color objects and on algorithms that track these objects.

The propagation of light underwater is complicated by several phenomena which affect both the illuminant and the light rays reflected from the object to the sensor. In particular, these are refraction, absorption and scattering [4]. *Refraction*, which causes light rays to bend while passing from one medium to another, is the reason for over– or under-estimation of depth. Waves and different salinity levels of sea-water also cause light rays to be bent at different angles even while passing through water. *Scattering* causes individual photons of light to be deflected or diverted, and is frequency dependant. Contrast between objects are greatly reduced underwater and also the transmission of color hues are influenced massively by scattering. *Absorption* of light is another common phenomenon underwater, as a large amount of light is lost with increased depth from the air-water surface. This phenomenon is also frequency dependant, which makes detection of certain colors difficult. These three phenomena uniquely alter behavior of light beams underwater, and hence influencing the way vision algorithms perform.

In section II, we discuss some background work regarding color tracking and the three algorithms we investigate. A brief discussion of the effects on light of open-ocean underwater environments also appears in that section. Section III describes the experimental setup used. Section IV gives the results of the tracking operations using the three approaches. We discuss with illustrations in Section V the implications of the results from Section IV, and attempt to characterize the salient features of underwater optics that can be exploited for enhanced visual tracking underwater. We conclude in Section VI by stating the future directions for this work and the ultimate objective of making our robot operate robustly using vision-dependent autonomy.

II. Color tracking

Using color features in visual tracking is an attractive option because of its simplicity and robustness under partial occlusion, depth and scale changes [5]. Nevertheless, there exist some significant problems that need to be addressed in order to design a robust and accurate color tracker. The biggest problem existing with color cues is color constancy

[6], which is defined as the removal of color bias due to effect of illumination. Issues like shadows, change in illumination and camera characteristics affect the phenomenon of color constancy. Since we are considering color trackers suitable for real-time applications such as UAVs, we seek a robust and efficient representation of the object colors, resulting in faster and accurate computation. The color space [5] [7] plays an important role in computational accuracy and robustness. We present the RGB and the HSV color spaces as a precursor to the tracking approaches we employ. Tracking algorithms used in the experiments all operate in the normalized-RGB (*i.e.* hue) space, which is obtained by dividing individual RGB values of each pixel by the sum of the values in the R,G and B channels. These tracking algorithms are also integrated into an on-line vehicle control system [8], as shown in Figure 1.

A. The RGB Color Space

The RGB(Red-Green-Blue) color space is a predominant representation for color representation. The RGB space uses a Cartesian coordinate system and forms a unit cube as shown in Figure 2.

In the normalized RGB space, each of the red, green and blue pixel values are divided by the sum of the RGB pixel values, such that the sum normalizes to one. The normalized RGB space is more stable to intensity and lighting variations, although there is an added overhead of transforming from the RGB to the normalized RGB space. We have also investigated the possibility of using the HSV color space, although further discussion of this color model is precluded due to space consideration.

B. Color Blob Tracking

The simplest approach to color based tracking is using a segmentation algorithm to detect objects of interest using their color features. The output of the segmentation algorithm is (possibly disconnected) regions in a binary image that match the color properties being tracked. These regions are termed ‘blobs’, and hence the approach is known as color blob tracking. We attempt to form these blobs through a *thresholding* process. By thresholding, we refer to the operation where pixels are turned ‘on’ if and only if their color values fall within a certain range and turned ‘off’ otherwise.

The blob tracker used for the experiments use the average normalized RGB values in a fixed-size window to set the low and high thresholds for the segmentation process.

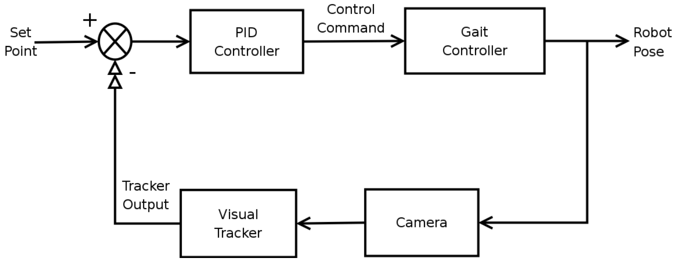


Fig. 1. Visual tracking integrated with vehicle control.

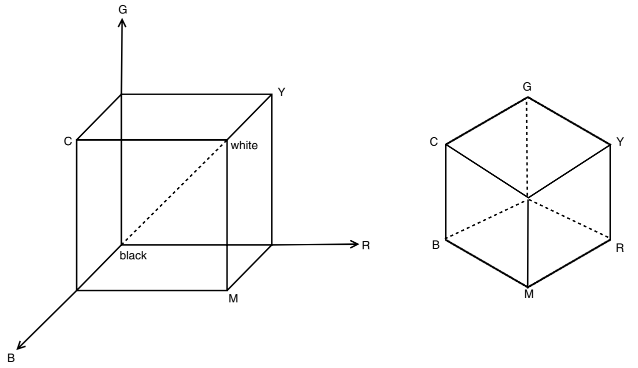


Fig. 2. The RGB color space.

C. Color Histogram Tracking

The color histogram [10] tracker works by first creating a color histogram of a fixed subregion of the image, presumably in the immediate neighborhood of the target to be tracked, which we refer to as the *target model histogram*. During the tracking stage, every incoming frame from the camera is divided into rectangular regions and their histograms are calculated. The similarities between the new *candidate* histogram and the target model histogram is calculated following one of several possible distance measures (to be discussed below). The subwindow with the highest match is chosen as the probable subwindow containing the target. The pattern of scanning the image for the target can be done sequentially, or in a spiral pattern starting from the location of target found in the previous frame. Depending on the application, the size and shape of the subwindow can also be made to change dynamically, although that makes the tracker computationally slightly expensive.

D. Mean-Shift Tracking

Mean-shift tracking [11] performs visual tracking by attempting to maximize the correlation between two statistical models of the underlying color distribution in the image. The correlation between the two distributions is expressed as a measurement derived from the Bhattacharyya coefficient [12]. Mean-shift trackers have been used to track objects based on color or texture, by building a statistical distribution of the feature being tracked. In effect, the mean-shift tracker relies on the *mean-shift vector* [13] to detect the direction of the change in gradient and correspondingly point to the (possible) new location of the target being tracked.

E. Distribution Similarity Measures

The tracking algorithms depend on a statistical measurement of similarity between two histograms to detect a possible match between the target and a candidate location. The following four methods of measurements are used in the experiments. We assume two histograms H and K, both having the same number of bins, N. The measurements compare h_i and k_j for

$i = j$, where h_i and k_j are the i-th and j-th bins of histograms H and K respectively.

1) Histogram Intersection Measure: The *histogram intersection similarity measurement* is calculated using the following formula:

$$d_{\cap}(H, K) = 1 - \frac{\sum_i min(h_i, k_i)}{\sum_i k_i} \tag{1}$$

This measurement has been proved useful in comparing histograms of different sizes [14].

2) The χ^2 (Chi-Squared) Measure: The χ^2 (chi-squared) metric is a measurement of the probability that one distribution was drawn from the other. The χ^2 measure is calculated by:

$$d_{\chi^2}(H, K) = 1 - \sum_i \frac{(h_i - m_i)^2}{m_i} \tag{2}$$

where,

$$m_i = \frac{h_i + k_i}{2}$$

The χ^2 metric does not permit the data in the underlying distributions to be percentages; they must be raw data. Also, the measured values have to be independent and observed frequencies must not be too small.

3) The Bhattacharyya Distance Measure: The Bhattacharyya coefficient has a direct geometric interpretation with respect to two distributions; for two m-dimensional unit vectors p and q, it is equal to the cosine of the angle between them. The Bhattacharyya distance between two histograms can be found using the following expression:

$$\rho_{Bhattacharyya}(H, K) = \sum_{i=1}^{m} \sqrt{k_i h_i} \tag{3}$$

It has been shown [13] that this measure is near-optimal and possesses scale invariant properties.

4) Jeffrey's Divergence: Jeffrey's Divergence has been derived from the Kullback-Leibler (K-L) divergence. The KL divergence measure is an information theoretic measure that can be interpreted as the inefficiency of transforming one distribution to the other using a code book. The KL measure, however is sensitive to quantization effects in the histogram computation (*i.e* bin size). Jeffrey's divergence is an empirically derived divergence that is numerically stable, insensitive to histogram binning and also robust in the presence of noise. The Jeffrey's divergence measure of similarity is calculated as follows:

$$\rho_J(H, K) = 1 - \sum_{u=1}^{m} (h_i log\frac{h_i}{m_i} + k_i log\frac{k_i}{m_i}) \tag{4}$$

where

$$m_i = \frac{h_i + k_i}{2} \tag{5}$$

III. Experimental Setup

We obtain footage from a controlled underwater environment of two targets of different color characteristics and run our tracking algorithms on this video footage. Lighting levels are varied and the water is disturbed to observe the effect of underwater currents. To create realistic underwater environments, we fill one-third of a glass tank of approximately $122.5cm \times 61.5cm \times 61.5cm$ with water. The bottom of the tank contains dead coral; for sediment, we use all-purpose sand for the bottom of the tank. The camera is placed at one end outside the tank, using a black lens hood to optically insulate the lens from receiving light from outside the tank. For each type of target, two sequences are recorded– one with low lighting and no water disturbance and one with increased lighting and disturbance. All three types of trackers are used on these four sequences, giving a total of 12 tracking sequences for the two targets. The number of frames of successful tracking are logged against the total number of frames in which the target is in the camera's field of view. We also obtain automatically the time span over which the trackers are able to successfully follow the target, since the frame-rate is constant throughout the sequences.

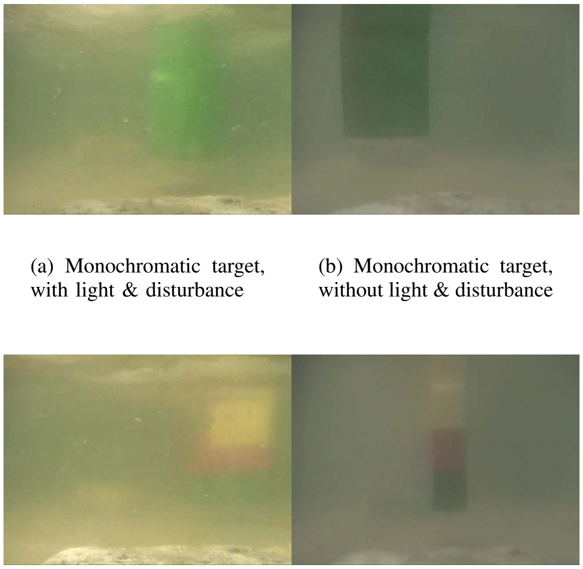
As mentioned before, the color blob tracker uses the average normalized RGB values in a 5×5 pixel window to set the low and high thresholds for the segmentation process. We use a median filtering algorithm [15] on a 5×5 window to remove 'salt-and-pepper' noise. The histogram tracker uses a 64-bin cumulative normalized histogram for storing the underlying color distribution. The mean-shift window used in our experiments has a 50 pixel diameter. We use a 3×32 histogram, *i.e.* a three-dimensional histogram with 32 bins per color channel for the target feature. To calculate histogram similarity, we use the Bhattacharyya distance measure described above.

Figure 3 shows one clip from each of these four sequences, showing the targets. The following section presents the results of tracking each of these frame sequences.

IV. Experimental Results

As stated, the goal of this experiment is to quantitatively analyze performances of tracking algorithms in underwater environments to provide a vision-based autonomous behavior for our underwater robot. We organize the results of the tracking algorithms below, based on tracking accuracy, target color characteristics and water and lighting conditions. Tracker outputs are compared manually with the ground truth obtained from the video sequences. We consider a successful tracking output to be only those frames in which the color blob tracker outputs a blob which contains part of the target, or the histogram or mean-shift trackers output a location on the image frame where a part of the target appears. Locking on to wrong objects (suspended particles, surface reflections) are also considered as mistracking. Table I shows the result of tracking a green cylindrical object– a monochromatic target.

Results of tracking an object comprised of blocks of three different colors–red, green and yellow, are listed below in Table II.



(a) Monochromatic target, with light & disturbance

(b) Monochromatic target, without light & disturbance

(c) Multichromatic target, with light & disturbance

(d) Multichromatic target, without light & disturbance

Fig. 3. Tracked objects in different water conditions

TABLE I
TRACKING RESULTS: MONOCHROMATIC TARGET

Tracker	Light/Waves	Tracked Frames	Missed Frames	Ratio
Color blob	Yes	372	234	61.39%
Histogram	Yes	181	390	31.70%
Mean-shift	Yes	208	366	36.24%
Color blob	No	800	10	98.77%
Histogram	No	697	113	86.05%
Mean-shift	No	441	416	51.46%

For comparison, we show the parameters for the color blob tracker for the green cylinder target both without and with lighting variations and water disturbance, in Table III and IV.

Figures 4 and 5 show the color histogram distribution for the single and multi-hued target objects, respectively. Note the distribution of color bins are spread out more for the multi-hued target.

V. DISCUSSION

Tables I and II give quantitative performances of each of the tracking algorithms. As expected, the presence or absence of disturbance and lighting variations affect each of the trackers considerably. For the monochromatic target, we see from the results that the color blob tracker performs the best, both with and without variations in water and lighting. This result is not unexpected, however, as the parameters of the blob tracker are tuned to track single color objects. If applied to track an object with a multitude of hues, the color blob tracker will output multiple blobs and will end up with many false positives.

Lighting variations affect the amount of hue transmitted through water, which in turn has an influence on the distance up to which the target can be tracked (or even, seen by the human eye) successfully underwater. For the monochromatic target, Table I shows that the successful tracking ratio is significantly lower with lighting variations and water disturbance than that of uniform lighting and calm water conditions. As mentioned above, the variations in lighting added with the water disturbance caused the target to blend in with the background, and subsequently the trackers failed to locate it at distances where they were successful previously. Figure 6 shows the multi-colored target under these two different conditions, along with the outputs from the histogram tracker. The output of the tracker is marked with a cross-hair outlined by a circle.

Performance of the color blob tracker under normal circumstances while tracking the multicolored target is worse than tracking under lighting variations. While not suitable for tracking multi-hued objects, the color blob tracker has been tuned on the yellow portion of the target block. With increased lighting levels, the amount of hue transmitted by the yellow block enable the blob tracker to lock on to the target in spite of disturbances in the water.

Both the histogram and mean-shift trackers performed poorly (31.7% and 36.24% success rate, respectively) tracking the single-colored cylinder. As one would expect, this result establishes the belief that a target object needs to have a sufficient mix of hues to be uniquely identified by the color distribution based trackers. The green cylinder, obviously lacked such variety in color and was henceforth mistracked a large number of frames by these two trackers.

Between the mean-shift and histogram trackers, we see from Table I and II that the histogram tracker performs significantly better than the mean-shift tracker on all but one instance. We attribute the better performance of the histogram tracker to its approach of performing a global search for the target. In contrast to the mean-shift tracker which performs a neighborhood search of the target object based on its previous location, the histogram tracker searches the entire input frame for a region that exhibits high similarity with the target histogram. This is a computationally more expensive approach than the one mean-shift uses, but this approach has been found to work very

TABLE II
TRACKING RESULTS: MULTICHROMATIC TARGET

Tracker	Light/Waves	Tracked Frames	Missed Frames	Ratio
Color blob	Yes	718	82	89.75%
Histogram	Yes	391	384	50.45%
Mean-shift	Yes	246	524	31.95%
Color blob	No	415	385	51.88%
Histogram	No	675	233	74.34%
Mean-shift	No	152	760	16.67%

TABLE III

BLOB TRACKER PARAMETERS; NO LIGHT/WATER DISTURBANCE

Color Channel	Low	High
Red	0.287317	0.327317
Green	0.38	0.42
Blue	0.272683	0.312683
Actual Color (63,82,60)		

TABLE IV

BLOB TRACKER PARAMETERS; WITH LIGHT/WATER DISTURBANCE

Color Channel	Low	High
Red	0.288157	0.328157
Green	0.412024	0.452024
Blue	0.239819	0.279819
Actual Color (102,143,86)		

well. The added computational overhead is compensated by the gains in tracking accuracy and robustness. It may be noted, however, that the search region of the mean-shift window can be expanded at an added computational cost, this in theory matching the performance of the histogram tracker.

One instance where the mean-shift tracker actually performs better than the histogram tracker, is during tracking the single-colored object with lighting variations and water disturbances. Since the histogram tracker works by looking at fixed-size rectangular windows, it is possible for the target to appear between two search windows, thereby confusing the tracker and losing track. The mean-shift tracker, on the other hand, does not change location in discrete steps. The green cylinder object tends to blend into the background with the light and water disturbances, and in this case we point out this phenomenon relating to the search windows as the reason for poor performance from the histogram tracker.

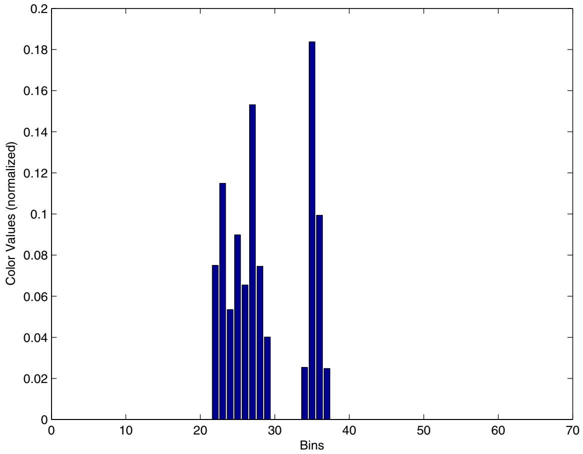


Fig. 4. Color Histogram for monochromatic target.

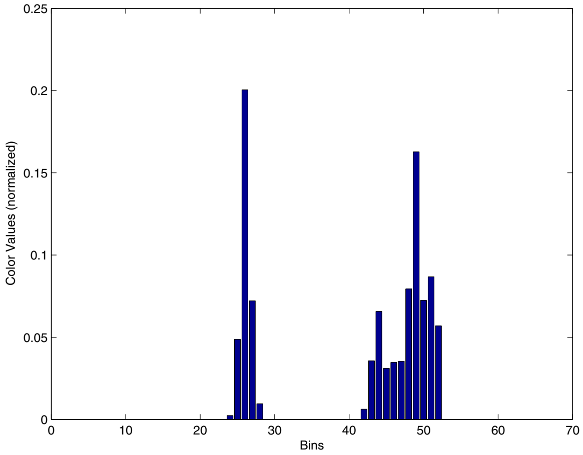
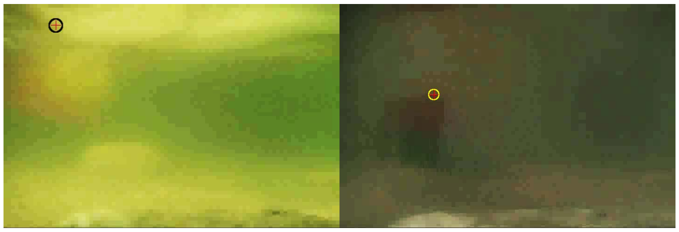


Fig. 5. Color Histogram for multichromatic target.



(a) With variations

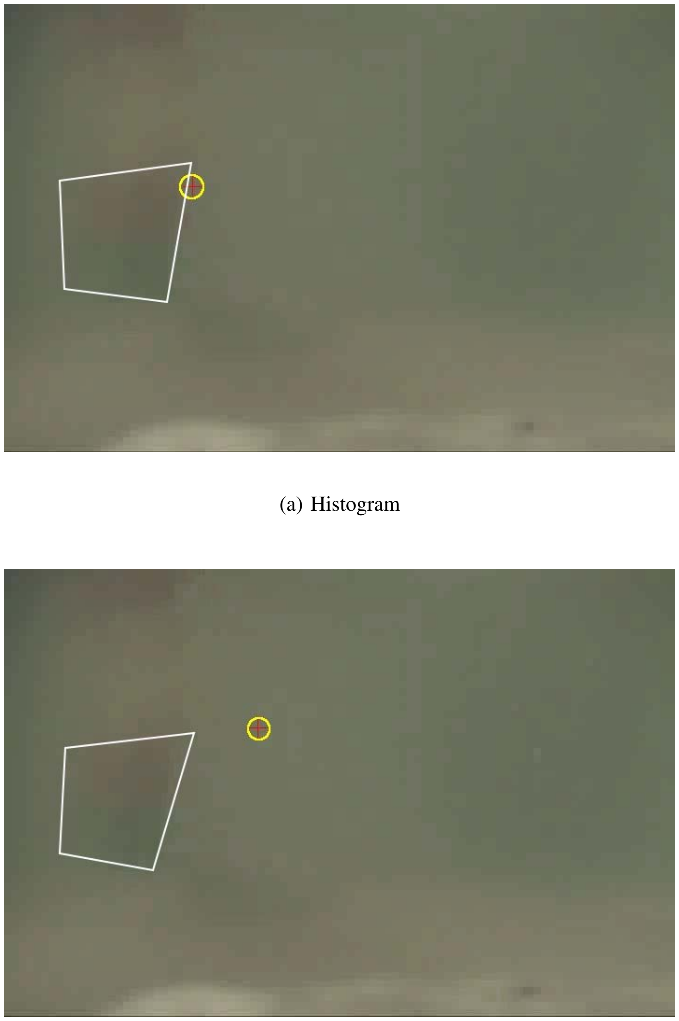
(b) Without variations

Fig. 6. Effect of distance and lighting variations on tracking the multichromatic target

Figure 7 shows the tracking results on one frame of the multi-colored object. The blob tracker failed to detect the target in this frame. The mean-shift tracker also failed to localize the target accurately. Only the histogram tracker was able to detect the target at the right location, even then barely locking on to the top-right corner. The target is outlined with a white border for clarity.

VI. CONCLUSION

We have presented a comparative discussion of three color tracking algorithms tracking objects with different color characteristics underwater. The key results are the success rates of the individual trackers under changing lighting conditions, disturbances in the water and motion of the targets. The approach was found to work well in field trials on our robot in both enclosed environments (a large pool) as well as in the open sea. We have been able to track a yellow ball 15cm in diameter using the color blob tracker over a distance of 27 meters on open-ocean trials on more than one occasion (the distance being limited by the fiber-optic tether length for the robot) [8], but those trials did not permit quantitative



(a) Histogram

(b) Mean-Shift

Fig. 7. Performance of histogram and mean-shift trackers on a colored object. The white trapezoidal region outlines the barely visible target. The yellow circle with the red cross-hair is where the trackers detect the target location. Both trackers fail to localize the target properly in this case, although the Histogram tracker performs somewhat better.

performance evaluation of the trackers. Color characteristics of the target objects were seen to be of major significance between the color blob and color distribution based trackers. We observed the effects of different underwater phenomena on lighting, and how it affects established vision algorithms that are known to work well in terrestrial (*i.e.* non-underwater) environments. The results summarize the performances of the three tracking algorithms and attempts to characterize the types of targets and environment conditions where each of the trackers would be best suited for tracking.

As we have discussed before, vision underwater is restricted by physical effects of the water medium on light rays. For a robust, efficient and effective tracking application, vision alone may not be sufficient, as the amount of light decreases exponentially with depth. Nevertheless, for shallow water applications, like monitoring marine life or servicing of marine equipment, vision can be a valuable sensing medium. To develop visually-guided underwater vehicles that strive for autonomy, development of algorithms suited for underwater vision are an absolute necessity. This study is a step towards identifying the problems vision applications face in underwater environments, and possible solutions for those problems without delving into the more complicated issues of novel algorithm design. In future work, the goal is to take these results and use them as the base for more robust, computationally efficient algorithms for underwater target tracking and apply these algorithms to make our underwater vehicle truly autonomous.

References

[1] L. Whitcomb, "Underwater robotics: Out of the research laboratory and into the field," in *IEEE International Confeference on Robotics and Automation*. IEEE, 2000, pp. 85–90.

[2] C. Georgiades, A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L. A. Torres-Mendez, P. Zhang, M. Buehler, G. Dudek, M. Jenkin, and E. Milios, "AQUA: An aquatic walking robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, IEEE. Sendai, Japan: IEEE Press, September 2004, pp. 3525–3531.

[3] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguère, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis, "A visually guided swimming robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005.

[4] T. T. Team, "The physics of diving: Light and vision," http://library.thinkquest.org/28170/35.html.

[5] V. Kravtchenko, "Tracking color objects in real time," Master's thesis, University of British Columbia, Vancouver, British Columbia, November 1999.

[6] G. D. Finlayson, "Computational color constancy," in *International Conference on Pattern Recognition*, vol. 1, Barcelona, Spain, September 2000, pp. 191–196.

[7] G. Klinker, S. Shafer, and T. Kanade, "A physical approach to color image understanding," *International Journal of Computer Vision*, vol. 4, no. 1. January 1990, pp. 7–38, January 1990.

[8] J. Sattar, P. Giguère, G. Dudek, and C. Prahacs, "A visual servoing system for an aquatic swimming robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005.

[9] A. R. Smith, "Color gamut transform pairs," in *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1978, pp. 12–19.

[10] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, November 1991.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–575, 2003.

[12] G. T. Toussaint and B. K. Bhattacharya, "Optimal algorithms for computing the minimum distance between two finite planar sets," *Pattern Recognition Letters*, vol. 2, pp. 79–82, 1983.

[13] D. Comaniciu and P. Meer, "Mean Shift: A robust approach toward feature space analysis," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[14] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[15] G. J. Y. T. S. Huang and G. Y. Tang, "Fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1979.