International Journal of Software Engineering and Its Applications
Vol.7, No.6 (2013), pp.459-468
http://dx.doi.org/10.14257/ijseia.2013.7.6.39

Comparative Studies of Remove Background algorithms for Objects Extraction of Underwater Images

Eun-Ju Kim and Sang-Soon Lee

leebone28@hanmail.net, sslee@gachon.ac.kr

Abstract

In this paper, two methods of extracting objects are compared through application to underwater images: one method is to extract objects by removing the background and quantifying it into a codebook by measuring the Mahalanobis distance for accurate object segmentation and extraction, and the other is to extract objects by removing the background and quantifying it into a codebook by measuring the Euclidean distance. In an experiment relating to the comparison and analysis, a standard underwater sample image was learned. Then, the background color's average value and the input image's stochastic distances were computed through the color similarity algorithm, and then the object was extracted after the background color could be removed. For the performance evaluation on the two algorithms, an underwater image was used to run some computer simulations. The experiment showed that an image applied with the color similarity algorithm had a better image segmentation performance than that with the different image technique.

Keywords: *Underwater Images, Mahalanobis Distance, Code Book, Image Segmentation*

1. Introduction

Related studies and researches have been very lack due to the difficult accessibility to an underwater image and the lack of underwater equipment. That being the case, such researches that sample an underwater image, apply an image processing algorithm and accordingly analyze the underwater image have been undertaken. In particular, this needs to be developed by using multiple algorithms after taking into account the certain particularities of the various environmental in underwater.

In recent years, many algorithms for color image segmentation have been studied. In existing studies on image segmentation, methods were researched to extract only the areas with similar characteristics to those of known objects from the input image by using the known object information [1-4]. In this paper, an algorithm for efficiently extracting objects from the underwater image is proposed. The proposed algorithm first removes the background color based on color similarities and then quantifies the remaining background color area, splitting the foreground from the background and extracting objects. The background is removed from the underwater image because it is efficient to remove the background area first and extract objects by teaching the underwater background image and comparing the similarities with a general underwater image as the background of an underwater image generally consists of blue colors. In Section 2 of this paper examines related work, and in Section 3 explains the suggested underwater image object extraction algorithm. In Section 4, the proposed algorithm is tested to evaluate its accuracy. Finally, Section 5 concludes.

ISSN: 1738-9984 IJSEIA

2. Related Work

Currently the amount of study on underwater image segmentation is insignificant. However, technologies to improve the quality of underwater images such as minimizing scattering effects by attaching a polarizing filter to the camera [5] or restoring images by using the distance information have been studied. While the SIFT algorithm shows results with poor performance to changes in turbidity, appearances in the underwater image can be somewhat discerned [6]. In addition, there are similar studies on extracting objects from the underwater image such as studies using underwater sonar radar or light attenuation and by Rajesh kumar Rai [7-9]. In the study by Rajesh kumar Rai, gray images are thickened enhancement by using CLAHE (contrast limited adaptive histogram equalization), and then splitting is done by applying the threshold value calculated from the histogram to the images. However, this algorithm has limitations to be applied to color images because measurement of color similarities is too simple due to gray images. In this paper, application of the algorithm to extract objects from general images to underwater images is tested. In particular, of the existing object extraction algorithms, the one to extract objects by removing the background was applied. This algorithm first teaches the background sample image and then extracts objects by measuring the pseudo distance to the learned background sample image. In this paper, the Mahalanobis and Euclidean distances were applied for comparison when measuring the pseudo distance.

3. The Design of Algorithms for Comparison

3.1. Underwater Background Learning

Unlike man-made artificial colors, underwater colors are mixture of various cyan and blue colors in harmony. They are distinguished from other objects and have certain ranges in RGB images. Therefore, color similarities can be detected using the RGB average and covariance in the underwater area by teaching only various background colors instead of objects. First, to get color similarity, teach in advance the RGB average and covariance for the standard underwater background color sample through equations (1), (2) and (3).

In equation (1), which is used to get the covariance of R and G, ($\overline{R}$) is the average value of R, n is the number of pixels in the learned underwater background color sample, and ($\sum RG$) is the covariance between R and G. In equation (2), which is used to covariance of G and B, ($\overline{G}$) is the average value of G, $n$is the number of pixels in the learned underwater background color sample, and ($\sum GB$) is the covariance between G and B. In equation (3), which is used to covariance of R and B, ($\overline{R}$) is the average value of R, nn is the number of pixels in the learned underwater background color sample, and ($\sum RB$) is the covariance between R and B [10].

$$\sum RG = \frac{1}{n}\sum_{i=1}^{n}(R_i - \overline{R_1})(G_i - \overline{G_1}) \quad (1)$$

$$\sum GB = \frac{1}{n}\sum_{i=1}^{n}(G_i - \overline{G_1})(B_i - \overline{B_1}) \quad (2)$$

$$\sum RB = \frac{1}{n}\sum_{i=1}^{n}(R_i - \overline{R_1})(B_i - \overline{B_1}) \quad (3)$$

Once the underwater background sample image is learned, matching background areas are removed by measuring similarities with the real-time image. In this paper, the Mahalanobis and Eucledean distances are used to measure similarities.

3.2. Similarity Measure-Mahalanobis Distances

Equations for calculating the Mahalanobis distance between the learned image and the real-time image are (4), (5) and (6). In these equations, LMR in LMR – RPi is the average of all R values of the learned image, and RPi means calculating the distances for R of the real-time image pixel by accepting one by one starting from 0. When this is divided by the standard deviation of the learned image, the resulting value by covariance is MD. Generalizing this, the Mahalanobis distance can be defined as follows.

$$MD_{Rp_i} = (LM_R - RP_i)L\sigma_R^{-1} \tag{4}$$

$$MD_{Gp_i} = (LM_G - GP_i)L\sigma_G^{-1} \tag{5}$$

$$MD_{Bp_i} = (LM_B - BP_i)L\sigma_B^{-1} \tag{6}$$

Similarities of the underwater background image calculated by equations (4), (5) and (6) can be derived byMD_{Rp}, MD_{Gp} andMD_{Bp}. The similarities between the colors of the image obtained in real time and those of the learned underwater background image can be measured by measuring the Mahalanobis distance between the actual object extraction target image and the underwater background image.

1. Selection of underwater background basis image
2. Learning mean, variance of underwater background basis image pixel
3. Comparing underwater background basis image
4. Set white pixel to background by Mahalanobis Distance
5. Select Mahalanobis Distance adapted image
6. Divide created code vector into two temporary code
7. Extracting object

Algorithm 1. Applied Mahalanobis Distance

```
1:  LOAD BI images(0,1,2,...,n);
2:  CALC m ← Mean(BI images(0,1,2,...,n));
        v ← Var(BI images(0,1,2,...,n));
        SAVE m, v, SQRT(v);
3: CALC mbi ← Mean(BI images(0,1,2,...,n));
        mri ← Mean(RI images(0,1,2,...,n));
    LOAD mbi, mri;
4:  LOAD m, v, SQRT(V);
    LOAD mbi(i), mri(j);
        FOR (i=0 ; i<n ; i++)
            FOR (j=0 ; j<n ; j++)
                MD(i) = (mbi(i) - mri(j))/SQRT(v);
                MD = MD + MD(i);
    CALC MDm ← Mean(M);
        SET MDm to Threshold;
            FOR (i=0 ; i<n ; i++)
                IF (mri(i) < MDm)
                SET mri(i) to (R=255 and G=255 and B=255);
5:  LOAD MD adapted image
6: LOAD Y0[16] = CodeBook[16];
```



```
        Dm+ = Y0(1+0.02);
        Dm- = Y0(1-0.02);
  7: Extracting object
     FOR (i=0 ; i<ε ; i++) (cf. ε=255)
      {
          FOR (j=0 ; j<16 ; j++)
        {
            FOR (k=0 ; k<n ; k++)
                IF(CB(j) ==ri(k))
                    SET ri(k) to (R=255 and G=255 and B=255);
        }
      }
```

3.2. Similarity Measure-Euclidean Distances

The Eucledean distance between the learned image and the real-time image is calculated by equation (7). The standard deviation can be calculated using the average and variance. Here, m is the average, and n is the number of pixels in the learned underwater image sample. Standard deviation is calculated for each R, G and B.

$$\sigma = \left[\frac{1}{n}\sum_{i=1}^{n}(X_i - m)^2\right]^{\frac{1}{2}} \tag{7}$$

In this paper, when the pixel value of the image where the color similarity is applied, it is checked whether there is a matching pixel in the generated codebook of the background image,and then objects are divided from the background. After removing the pixel in the matched background, it moves to the next pixel. Then the codebook is updated and the steps are repeated.

1. Selection of underwater background basis image
2. Learning mean, variance of underwater background basis image pixel
3. Comparing underwater background basis image
4. Set white pixel to background by Euclidean Distance.
5. Select Euclidean Distance adapted image
6. Divide created code vector into two temporary code
7. Extracting object

Algorithm 2. Applied Euclidean Distance

```
1:  LOAD BI images(0,1,2,...,n);
2:  CALC m ← Mean(BI images(0,1,2,...,n));
      v ← Var(BI images(0,1,2,...,n));
      SAVE m, v, SQRT(v);
3: CALC mbi ← Mean(BI images(0,1,2,...,n));
      mri ← Mean(RI images(0,1,2,...,n));
   LOAD mbi, mri;
4:  FOR (i=0 ; i<n ; i++)
        FOR (j=0 ; j<n ; j++)
              STORE Temp ← SQRT(ABS(mri(i) − mbi(j)));
              IF (MIN(Temp)) SELECT(mbi(j), mri(i));
              ELSE NEXT;
```

```
        LOAD mbi ← mbi(j),  mri ← mri(i);
    CALC T_ED ← ABS(mri − mbi);
    LOAD mri(i);
    FOR (i=0 ; i<n ; i++)
            IF(mri - T_ED <= mri(i) <= mri + T_ED)
                mri(i) = 0;
5:  LOAD ED adapted image
6: LOAD Y0[16] = CodeBook[16];
      Dm+ = Y0(1+0.02);
      Dm- = Y0(1-0.02);
7: Extracting object
     FOR (i=0 ; i<ε ; i++) (cf. ε=255)
     {
          FOR (j=0 ; j<16 ; j++)
       {
             FOR (k=0 ; k<n ; k++)
                 IF(CB(j) ==ri(k))
                     SET ri(k) to (R=255 and G=255 and B=255);
       }
}
```

4. Experiments

4.1. Experimental Methods

For test evaluation, the amount of removed background colors is measured. Here, the amount of colors removed from the desired object is measured simultaneously to compare the error rate and the success rate. In this paper, the method of comparing pixels is used to analyze the accuracy of the test results [11].

The detailed description of the analysis method is as follows:

ⅰ)Manually separate the original image into the background and objects.

ⅱ)Manually count the number of pixels in the completed background area and the object area.

ⅲ)Calculate the SUCCESS rate with equation (8) to figure out how accurately similar pixels were found when removing the background area.

$$\text{SUCCESS (\%)} = \left(\frac{P_{bg}}{P_{bgtot}}\right) \times 100 \tag{8}$$

where P_{bg} is the number of removed pixels in the background except the objects, and P_{bgtot} is the total number of removed pixels in the background except the objects.

ⅳ)Calculate the ERROR rate with equation (9) to figure out how much of the object area was removed.

$$\text{ERROR (\%)} = \left(\frac{P_{obj}}{P_{objtot}}\right) \times 100 \tag{9}$$

where P_{obj} is the number of removed pixels in the object, and P_{objtot} is the total number of removed pixels in the object. The ERROR rate can be high even when the SUCCESS rate is 100%, and the SUCCESS rate can be low even when the ERROR rate is 0% if the background is not completed removed. Therefore, both rates should be considered in all evaluations. Equation (10) was used to express the accuracy of the segmentation as a single number.

$$\text{ACCURACY (\%)} = \left(\frac{SUCCESS-ERROR}{SUCCESS+ERROR}\right) \times 100 \tag{10}$$

In equation (10), which is used to calculate the accuracy, the total removal is the sum of successful background removal and failed object removal. Here, the accuracy can be calculated by subtracting failed object removal from the successful background removal. That is, the sum of background removal object removal comes as the denominator, and the ACCURACY for success can be obtained by subtracting failure from success.

4.2. Results

To test the algorithm suggested in this paper, the results were derived by applying two algorithms using the custom-implemented application. Three original images (768*768) were used for test data. In Figure 1, the first image is defined as image#1, the second image is defined as image#2, and the third image is defined as image#3. ED is Euclidean Distance, and MD is Mahalanobis Distance.

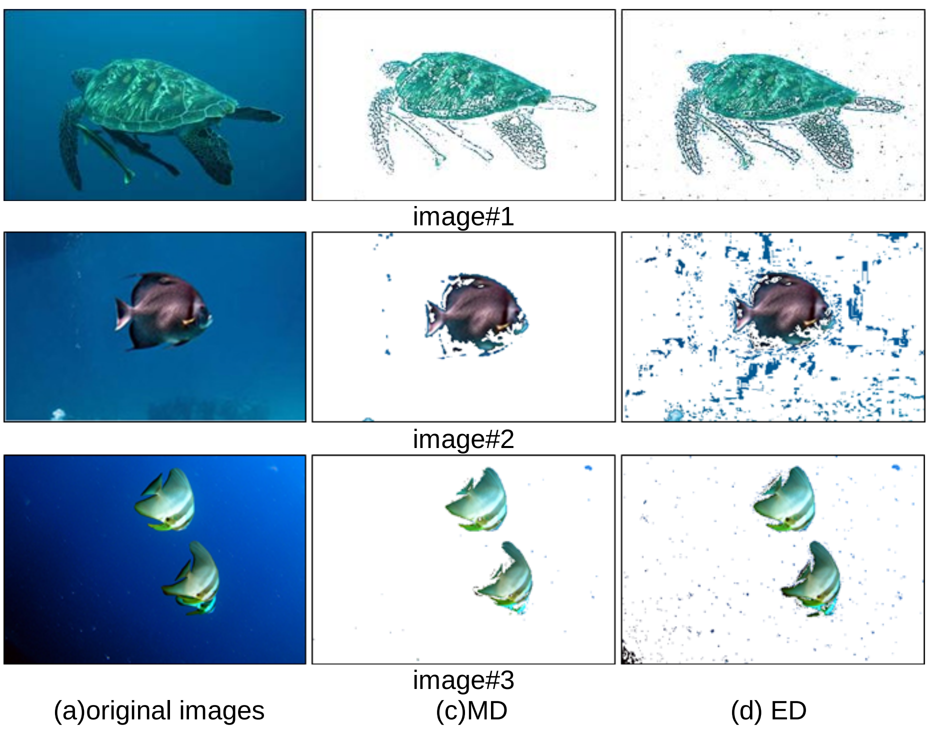


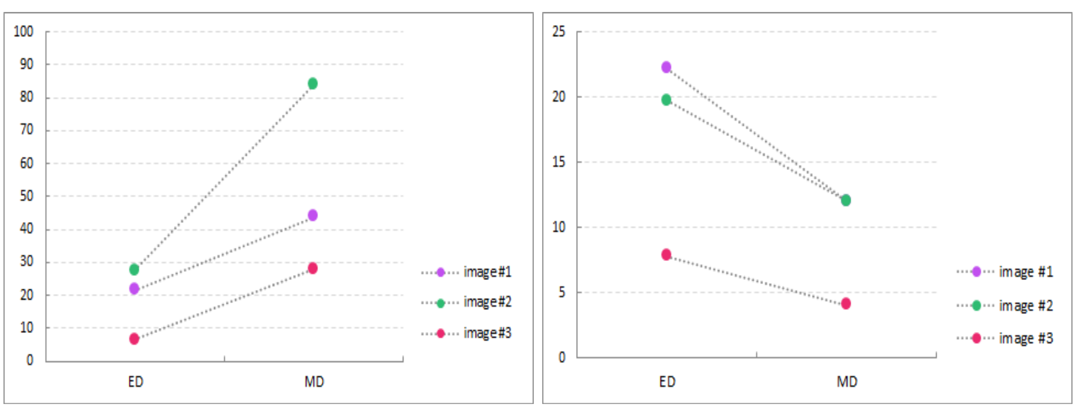
Figure 1. Result of Images

The results are assessed by the pixel count to figure out how much background colors were removed and how well the desired objects were preserved. Here, S is an abbreviation of SUCCESS and E is an abbreviation of ERROR. In Table 1, S is the ratio of the white (removed) pixels to the total pixels, and E is ratio of the white (removed) pixels to the object pixels. That is, the higher white ratio of SUCCESS, the better the background was removed, and the lower white ratio of ERROER, the less loss to the object. How the ratios change depending on the algorithm of each image, not the phase, is important because the height of the graph is determined by the ratio of pixels that occupy the image.

Table 1. Removal Rate of Background

Pixel rate / algorithm	image #1		image #2		image #3	
	S	E	S	E	S	E
ED	21.49	22.16	27.62	19.74	6.61	7.82
MD	43.89	12.03	84.05	12.03	47.88	4.06

Table 1 shows the ratio of removed pixels for each image. The SUCCESS rate of the ED for each image was measured as 43.89% for image#1, 84.05% for image#2 and 47.88% for image#3. The SUCCESS rate of the MD was measured as 43.89% for image#1, 84.05% for image#2 and 47.88% for image#3. The ERROR rate of the ED was measured as 22.16% for image#1, 19.74% for image#2 and 7.82% for image#3. The ERROR rate of the MD was measure as 12.03% for image#1, 12.03% for image#2 and 4.06% for image#3. These results indicate that the MD extracts objects more accurately from underwater images where specialty is reflected because the MD measures the similarities more accurately than the ED. As shown in Figure 3, the MD has the higher SUCCESS rate and the lower ERROR rate than the ED.



(a) SUCCESS (b) ERROR

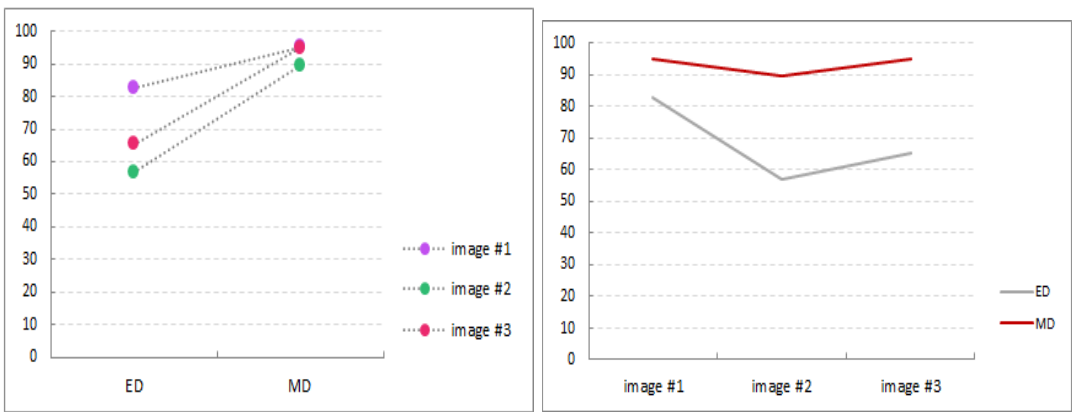
Figure 2. The Graph for Removal Rate

Then the ACCURACY was calculated by equation (10). As shown in Table 2, for the case where the ED was applied, 82.62% was measured for image#1, 56.75% for image#2 and

65.20% for image#3. For the case where the MD was applied, 95.07% was measured for image#1, 89.53% for image#2 and 97.02% for image#3. Figure 4, which shows the graph of ACCURACY, shows that the MD is more accurate than the ED. These results confirms that the proposed algorithm extracts objects from underwater images better than existing algorithms.

Table 2. Accuracy Rate

image / algorithm	image #1	image #2	image #3
ED	82.62	56.75	65.20
MD	95.07	89.53	97.02



(a) The graph for algorithms (b) The graph for images

Figure 3. ACCURACY

5. Conclusion

Underwater images are a snapshot of a changing environment that is very difficult to predict due to the presence of several factors, including strong and elements suspended matter, turbidity, aquatic organisms, and ocean currents. This leads to distorted underwater video. Existing object extraction methods using the image segmentation algorithm have proven difficult, as underwater images are also distorted. Therefore, In this paper, two methods of extracting objects are compared through application to underwater images: one method is to extract objects by removing the background and quantifying it into a codebook by measuring the Mahalanobis distance for accurate object segmentation and extraction, and the other is to extract objects by removing the background and quantifying it into a codebook by measuring the Eucledean distance. According to the results of the experiment, the accuracy of the Euclidean distance was 68%, while the accuracy of the Mahalanobis distance was measured at 93%. The experiment showed that the Mahalanobis distance resulted in better image segmentation performance when compared with the Euclidean distance. As a result of this study, further research into various algorithms for underwater image object extraction is needed to develop the system so that it can be applied universally to various water environments.

References

[1] J. Cheng, S. Drue, G. Hartmann and J. Thiem, “Efficient Detection and Extraction of Color Objects from Complex Scenes”, IEEE Pattern Recognition, vol. 1, no. 11, **(2000)**, pp. 668-671.
[2] R. Schettini, “Multicolored object recognition and location”, Parrern Recognition Lett, vol. 15, **(1994)**, pp. 1089-1097.
[3] F. Ennesser and G. Medioni, “Finding Waldo, or focus of attention using local color information”, IEEE Pattern Analysis & Machine Intelligence, vol. 17, no. 8, **(1995)**, pp. 730-742.
[4] V. V. Vinod and H. Murase, “Focused color intersection with efficient searching for object extraction”, Pattern Recognition, vol. 30, no. 10, **(1997)**, pp. 1787-1797.
[5] Y. Y. Schechner and N. Karpel, “Clear underwater vision”, Proc. Computer Vision & Pattern Recognition, vol. 1, **(2004)**, pp. 536-543.
[6] N. Carlevaris-Bianco, A. Mohan and R. M. Eustice, “Initial results in underwater single image dehazing”, University of Michigan, **(2010).**
[7] K. Oliver, W. Hou and S. Wang, “Image feature detection and matching in underwater conditions”, Progress in Biomedical Optics and Imaging, vol. 7, **(2010)**, pp. 7678.
[8] S. P. Bazeille, L. Quide and L. Jaulin, “Color-based underwater object recognition using water light attenuation”, Intelligent serivce robotics, vol. 5, no. 2, **(2012)**, pp. 109-118.
[9] R. kumar Rai, P. Gour and B. Singh, “Underwater Image Segmentation using CLAHE Enhancement and Thresholding”, International Journal of Emerging Technology and Advanced Engineering, vol. 2, **(2012)**, pp. 118-123.
[10] K. Eun-Ju and L. Byung-Soo, “Comparison and Analysis of Algorithms for Object Segmentation of Underwater Images”, Korean Institute of Information Technology, vol. 11, no. 5, **(2013)**, pp. 203-210.
[11] R. A. Zoroofi, Y. Sato, T. Sasama, T. Nishii, N. Sugano, K. Yonenobu, H. Yoshikawa, T Ochi and S. Tamura, “Automated Segmentation of Acetabulum and Femoral Head From 3-D CT Images”, IEEE Transactions on Information Technology in Biomedichine, vol. 7, no. 4, **(2003)** December, pp. 329-343.

Authors

Eun-Ju Kim, received her MS in computer science and engineering from Incheon National University, Korea, in 2010. She is currently working toward a PhD in computer science and engineering at the same university. Her research interests include computer vision, image processing, and Artificial intelligence.

Sang-Soon Lee, received his MS in computer science and engineering Inha University, Korea, in 1986. Received his PhD in computer science and engineering from Incheon National University, Korea, in 2005. He is a professor in the Department of Computer Science and Engineering, Gachon University, Korea. His current research interests include automata, system software, micro-processor, RFID/USN, and IT convergence.

