DETECTION OF MATCHINGS IN A SEQUENCE OF UNDERWATER IMAGES THROUGH TEXTURE ANALYSIS

Rafael Garcia, Xevi Cufi and Joan Batlle

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona, E.P.S.
17071 Girona, Spain
e-mail: {rafa,xcuf,jbatlle}@eia.udg.es

ABSTRACT

This paper presents an approach to ameliorate the reliability of the correspondence points relating two consecutive images of a sequence. The images are especially difficult to handle, since they have been acquired by a camera looking at the sea floor while carried by an underwater robot. Underwater images are usually difficult to process due to light absorption, changing image radiance and lack of well-defined features. A new approach based on gray-level region matching and selective texture analysis significantly improves the matching reliability.

1. INTRODUCTION

Intensity-based region-correlation techniques have been extensively used to search for correspondences between pairs of images [1,2]. Although these approaches lead to successful matchings in well-contrasted images, in some cases the lack of image features cause the matching procedure to fail. The problem we aim to solve is the detection of matches in consecutive images of a sequence, in order to estimate the motion of an underwater vehicle. Likewise, we want to construct a visual map of the zone surveyed by our submersible [3,4,5]. For this purpose, the vehicle is equipped with a color camera which acquires images of the bottom of the sea. The most crucial step in the estimation of the vehicle motion consists on detecting matchings between image pairs. Unfortunately, underwater images are difficult to process due to the medium transmission properties [6]. These difficulties provoke a blurring of the elements of the image with high clutter in the regions of interest and lack of distinct features. Moreover, the limited range forces the vehicle to carry artificial light, introducing new properties to the image, such as low contrast and non-uniform illumination.

2. TEXTURE ANALYSIS

We have tested the behavior of different texture operators in underwater images. Concretely, we searched for texture parameters that remain constant for the same scene patch for the whole image sequence. One of the operators that have been used are the *texture energy filters* [7], which are derived from the computation of a series of statistical measures (mean and standard deviation) on a pre-filtered image. This pre-filtered image is obtained by applying a set of masks (3×3 or 5×5) that define some textural properties of the image. In order to obtain these masks, a series of vectors defining some textural proprieties are combined. The typical vectors are *level*, *edge*, *spot*, *wave*, *ripple* and *oscillation*. See [7] for further details.

A second texture operator based on the spatial distribution of pixels in the image has been used: *Co-occurrence matrix* [8]. It takes into account the frequency of appearance of the pairs of pixels located at a distance d and an angle θ (co-occurrences). A set of statistics can be computed for every co-occurrence matrix, obtaining the textural characteristics of the image. We have selected *contrast* and *correlation* as the most discriminating statistics to be used.

Finally, since a textured region can be described by means of its texture spectrum –that is, a set of values called texture units– a set of 3×3, 5×5 and 7×7 simple local patterns can be defined. The different texture units can be determined from these patterns, obtaining a texture measure of the considered region. The contrast measures provided by these last texture operators (*Local Binary Patterns* [9]) have also been used in our study.

It should be taken into account that the first two operators can generate several measurements, depending on the number of orientation angles, the distance of correlation and the size of the neighborhood. We have tested several configurations of these texture operators in order to find the most advantageous set-up for our application.

Finally, we chose 4 different angles for the coocurrence matrix, taking only distances of 1-pixel, and 9 masks of the energy filter taking only a 3×3 neighborhood.

3. FINDING CORRESPONDENCES

The matching algorithm is sequenced as following: first, the high gradient areas of the first image are detected through a detector of interest points (some sort of corner detector). Then, for every corner in the first image, a set of possible matchings is established in the second image. These matches are detected by means of area-correlation. Finally, a set of texture measures is taken for the area surrounding the original corner in the first image, and this texture set is compared to the textures computed at every possible matching on the second image. As will be shown in the results, the most similar texture patch corresponds normally to the correct match.

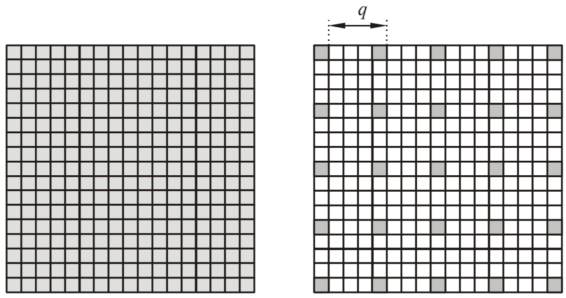
3.1. Detection of interest points

A very simple and fast detector of interest points has been implemented. First, a Canny edge detector is applied to the image [10], binarizing the output of the filter at a quite high threshold. Thus, an undersegmented image containing only the most relevant contours of the image is obtained. Next, a cross mask is applied to only the edge pixels that are left. A pixel is considered to be an *interest point* if it is in the intersection between two straight lines, that is to say, if it has 3 or more neighbors also belonging to any edge.

3.2. Gray-level region-correlation

In order to establish correspondences between images a classical correlation technique can be applied to the intensity images. However, since our images are acquired by a color camera, we have found that in some cases the correlation produces better results in the blue band of the image. This fact is related to the variation of the optical properties of different water bodies depending on the interaction between the light and the aquatic environment [6]. Given that the light suffers less absorption when it has a higher frequency, the blue component of the image provides higher contrast than the average of all frequencies, that is, the intensity component. Before correlation, our images are low-pass filtered with a Gaussian mask in order to reduce the inherent acquisition noise.

As the amount of interest points increases, the classic correlation approach is very time consuming. For this reason, we use a subsampled version of the correlation window. This means that if the correlation window has a size of $C \times C$ pixels, then only every q[th] pixel of the window is taken into account, reducing the processed pixels to a $C' \times C'$ matrix, where $C' = ((C-1)/q)+1$. Figure 1 illustrates an example with C=17 and q=4. The accuracy of reducing the amount of data in the correlation window is practically the same as using the full window [2]. This is due to the strong correlation in the gray level of neighboring pixels, producing smooth intensity variations, especially after the low-pass filtering.



17×17 pixels are considered   5×5 pixels are considered

Figure 1. Data reduction of a correlation window of 17×17 pixels. Only one pixel every four is taken into account to compute correlation.

Given an interest point p_1 of the first image, with coordinates (x_1,y_1), the subsampled correlation window is centered at this point. Then, a search window is defined on the second image, also centered at (x_1,y_1), and the correlation operation is performed. A correlation score is computed in the following way [11]:

$$corr(p_1,p_2) = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=\alpha}^{\alpha}\left(I_1(x_1+i\cdot q, y_1+j\cdot q)-\overline{I_1(x_1,y_1)}\right)\cdot\left(I_2(x_2+i\cdot q, y_2+j\cdot q)-\overline{I_2(x_2,y_2)}\right)}{\alpha^2\sqrt{\sigma^2(I_1)\cdot\sigma^2(I_2)}} \tag{1}$$

where $\alpha = (C-1)/2q$; I_1 and I_2 are the first and second images, respectively; $\sigma^2(I)$ is the variance of the image computed in the correlation window (see equation 2); and $\overline{I(x,y)}$ is the average of the correlation window in the image as shown in equation 3.

$$\sigma^2(I) = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=-\alpha}^{\alpha} I(x+i\cdot q, y+j\cdot q)^2}{\alpha^2} - \overline{I(x,y)}^2 \tag{2}$$

$$\overline{I(x,y)} = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=-\alpha}^{\alpha} I(x+i\cdot q, y+j\cdot q)}{\alpha^2} \tag{3}$$

We then define a minimum threshold for a matching point to be considered a possible correspondence of a given interest point. In this way, for each point in the first image, we thus have a set of p candidate matches in the second image. The number p of possible matches may be different from one interest point to another.

3.3. Texture extraction and similarity measure

Given an interest point, the problem now is to decide which is the right match among the p candidates selected by the correlation procedure. A 7×7 neighborhood is selected around the interest point, and it is subsampled with q=3, obtaining a 3×3 window. The texture operators defined in section 2 are computed for the points in the subsampled window. In this way a texture vector is obtained for the neighborhood of the interest point. Every point of the neighborhood provides 9 measures of the energy filters, 8 of the co-occurrence matrix (4 orientations and 2 statistical measures), and 3 of the Local Binary Pattern operator. Thus, the texture vector contains $3^2\times(9+8+3)=180$ texture values. The same operation is performed in the second image centering the 7×7 window on every one of the p possible matches. Then, the p texture vectors are compared with the texture vector of the interest point by computing the point-to-point Euclidean distance. The best match is selected as the one minimizing the following distance:

$$d(\vec{a},\vec{b}_j) = +\sqrt{\sum_{i=1}^{180}(a_i - b_i)^2}, \qquad \forall j \in [1..p] \qquad (4)$$

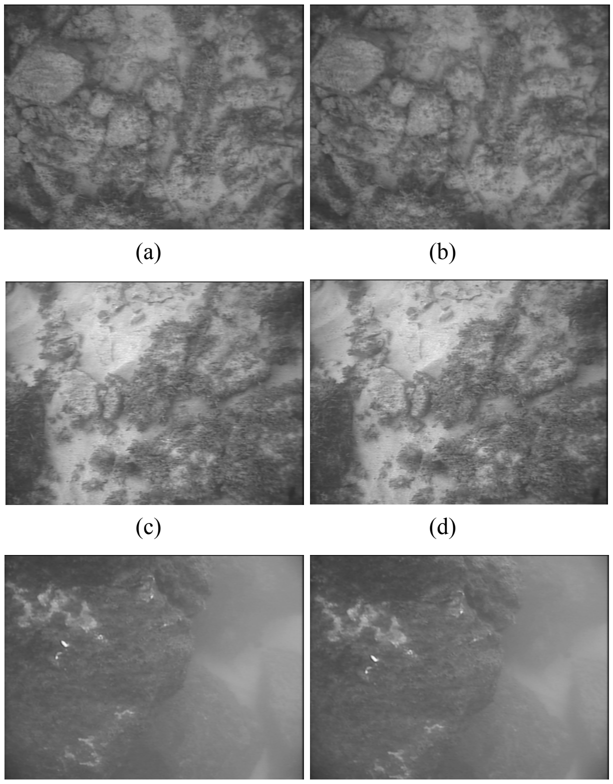
where $\vec{a}$ is the texture vector of the interest point in the first image, and $\vec{b}_j$ stores the texture attributes of every candidate matching.

4. RESULTS

Several experiments have been performed in order to validate the texture-based matching strategy. Three typical underwater situations have been used to test our system. The images have been acquired by a color camera carried by an underwater robot while this was being teleoperated. The acquisition frame rate was set to 3 f.p.s. In the first sequence, figures 2a and 2b, the vehicle is moving forward on a typical surveying mission. In spite of the fact that image features are available through the entire image, the low contrast of the image makes feature extraction a difficult task. The second sequence, figures 2c and 2d, shows the effect of a turning. Fortunately, the robot is not able to perform this motion at a high speed, producing a smooth movement that draws to a small image rotation. In the last sequence the vehicle is moving backwards while an operator is trying to keep station. Note the lack of image contrast in the right and the center of the image.

In order to evaluate the accuracy of the matching procedure we have used a two step approach. First, the interest points have been detected in the first image, and the region correlation has been applied to the second image. Next, a human operator has marked all the visually incorrect matches, in order to know which correspondences have been incorrectly established.

Figure 3 shows the result of the automatic detection of matches by using the classical correlation procedure (left column), and the effect of applying the texture analysis to all the candidate matches. In the first sequence 205 interest points were detected. After the region correlation procedure 9 points were incorrectly matched. The texture analysis was then applied to all the candidate matches, producing 5 false matches. Among them, 4 belonged to the set of correlation mismatches, while the fifth one was a correct match in the correlation procedure, but an incorrect match in the texture analysis. The second sequence is composed by 150 points. 8 wrong matches were produced by correlation. The texture analysis successfully matched 149 points. This high rate is due to the existence of clearly differentiated zones in the image with rather different textures, and the small motion performed by the vehicle. Finally, in the last sequence 69 interest points were detected with 4 incorrect matches through correlation. The texture analysis corrected these 4 matches, but introduced a new false match among the others.



(a) (b)

(c) (d)

(e) (f)

Figure 2. Sample images of the tested sequences.

5. CONCLUSIONS AND FURTHER WORK

We have proposed in this paper a method to improve image matching in underwater image sequences. The accuracy of the method is directly related to the exploitation of the local texture parameters in the image. Although the presented technique has been tested in underwater imaging, it can be applied to most of the terrestrial applications where poor image contrast require of a wide study of the textural properties of the features to match.

Our approach has proved to perform better than the merely use of classical region correlation. However, in some situations, it introduces new false matches at the same time as increases the computational cost of the matching procedure. For this reason further efforts should be devoted to explore the possibilities of other texture operators to characterize the interest points. New statistical measures can be applied to the pre-filtered images, so as to obtain a more accurate region characterization. On the other hand, a robust estimation technique (*i.e.* LMedS) can be applied after the texture analysis, ameliorating the quality of the correspondences.

6. REFERENCES

[1] D.M. Mount, N.S. Netanyahu, J. Le Moigne, "Efficient algorithms for robust feature matching", *Pattern Recognition*, no. 32, pp. 17–38, 1999.

[2] A. Giachetti, "Matching techniques to compute image motion", *Image and Vision Computing*, no. 18, pp. 247–260, 2000.

[3] R. Garcia, J. Batlle, X. Cufí, and J. Amat, "Positioning an Underwater Vehicle through Image Mosaicking", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.

[4] Xu, X., Negahdaripour, S., "Vision-based motion sensing from underwater navigation and mosaicing of ocean floor images", in *Proc. of the MTS/IEEE OCEANS Conference*, vol.2, pp. 1412–1417, 1997.

[5] N. Gracias and J. Santos-Victor, "Underwater Video Mosaics as Visual Navigation Maps", *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, 2000.

[6] C. J. Funk, S.B. Bryant, P.J. Beckman Jr., "Handbook of underwater imaging system design", Ocean Technology Department, Naval Undersea Center, 1972.

[7] K.I. Laws, "Textured Image Segmentation", Ph.D. *Thesis*, Processing Institute, University of Southern California, Los Angeles, 1980.

[8] R.M. Haralick, K. Shanguman, and I. Dinstein, "Textural Features for image classification", *IEEE Trans. on Systems, Man and Cybernetics*, vol. 3, pp. 610–621, 1973.

[9] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative Study of Texture Measures with Classification Based on Feature Distribution", *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

[10] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[11] R. Klette, K. Schlüns, A. Koschan, "Computer Vision: Three-Dimensional Data from Images", Springer-Verlag, 1998.

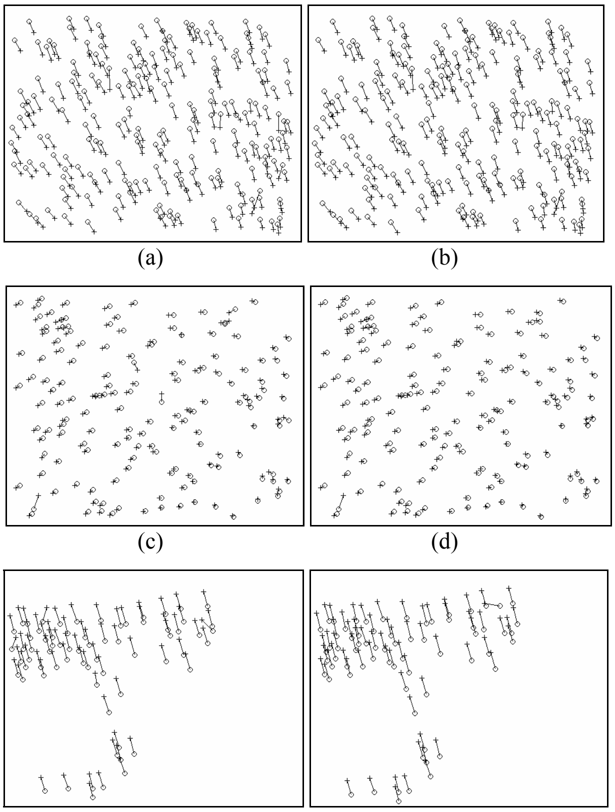


Figure 3. Correspondences computed from the images of figure 2. The circles correspond to the interest points, while the crosses are their matchings in the next image. (a), (c) and (e) illustrate the gray-level correlation results. The interest point is directly matched to the correspondence with the higher correlation score. (b), (d) and (f) show the result of the texture matching.