Positioning an Underwater Vehicle through Image Mosaicking

R. Garcia, J. Batlle, X. Cufi

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

J. Amat

Dept. of Automatic Control and Computer Eng.
Polytechnical University of Catalonia
08028 Barcelona, Spain
e-mail: amat@esaii.upc.es

Abstract

Mosaics have been commonly used as visual maps for undersea exploration and navigation. The position and orientation of an underwater vehicle can be calculated by integrating the apparent motion of the images which form the mosaic. A feature-based mosaicking method is proposed in this paper. The creation of the mosaic is accomplished in four stages: feature selection and matching, detection of points describing the dominant motion, homography computation and mosaic construction. In this work we demonstrate that the use of color and textures as discriminative properties of the image can improve, to a large extent, the accuracy of the constructed mosaic.

The system is able to provide 3D metric information concerning the vehicle motion using the knowledge of the intrinsic parameters of the camera while integrating the measurements of an ultrasonic sensor. The experimental results on real images have been tested on the GARBI underwater vehicle.

1 Introduction

The task of positioning an underwater vehicle can take advantage of the rich amount of information available at the bottom of the sea when viewed from a camera. The construction of a composite image that combines the set of frames taken from the submersible can greatly help in this task. This image is known in the literature as a *mosaic*, and is commonly used as a visual map for undersea exploration and navigation. In order to construct ocean floor mosaics, the individual images forming the mosaic are usually obtained by setting a camera on a ROV or AUV. The camera looks down, parallel to the bed of the sea, and the acquired images cover a small area of the ocean floor. In this way, the position and orientation of the underwater vehicle can be calculated by integrating the motions from one image to the next [1,2].

Unfortunately, underwater images often lack distinct features that are commonly exploited in terrestrial vision systems for detecting motion. Moreover, the range is limited and the need for artificial light introduces many new properties to the image, such as low contrast, non-uniform illumination and scattering. Quite often, small particles suspended in the water show up as marine snow making difficult the feature extraction and matching processes.

One of the first computer-aided systems to automate the construction of underwater mosaics was presented by Haywood in [3]. In this work, no feature extraction was performed at all, and mosaicking was accomplished by snapping images at well-known positional coordinates, and warping them together since the registration was known beforehand. Marks, *et al.* developed a completely autonomous column-based mosaicking system in [4] by using a constrained four-parameter semi-rigid motion model. Some years later, unconstrained image mosaicking was obtained by applying smoother-follower techniques to reduce image alignment errors within the mosaic [5]. In both cases, the registration between images was computed by correlating binary images, after going through a *signum of Laplacian of Gaussian* filtering process, which attenuated the effect of nonuniform illumination. Negahdaripour, *et al.* have detected motion from seabed images through recursive estimation of optical flow [6]. They studied this problem in the presence of intensity variations and underwater medium effects [7], and developed a "Direct Method" for motion estimation [8]. This direct estimation of motion has been successfully applied to mosaicking (*e.g.*, [1]) and station-keeping [9]. However, the application of gradient-based techniques is not always accurate in low contrast environments. Other works in underwater mosaicking have made use of image corners and gray-level pixel-correlation to detect correspondences [2], achieving successful and accurate underwater mosaics in well-contrasted images.

To our knowledge, none of the works described above has addressed the problem of feature characterization as a whole, in order to improve the correspondences between images. We propose a method to solve the matching problem by means of a wide study of texture and color. A texture-based mosaicking method is proposed in this paper in order to estimate the position of the GARBI underwater submersible [10].

The paper is organized as follows: first, a brief description of the GARBI Underwater Vehicle is given. Next, the main algorithm to robustly construct a mosaic in a low-contrast scenario is detailed in section 3. Finally, the following sections present some of the results obtained in

a sea mission with GARBI, and summarize the conclusions of our work.

2 The GARBI Underwater Vehicle

GARBI [10,11] was first conceived as a *Remotely Operated Vehicle* (ROV) for exploration in waters up to 200 meters in depth. At the moment, a control architecture is being implemented to transform this vehicle into an *Autonomous Underwater Vehicle*. GARBI (see Figure 1) was designed with the aim of building an underwater vehicle using low cost materials, such as fiber-glass and epoxy resins. To solve the problem of resistance to underwater pressure, the vehicle is servo-pressurized to the external pressure by using a compressed air bottle, like those used in scuba diving. Air consumption is required only in the vertical displacements during which the decompression valves release the required amount of air to maintain the vehicle's internal pressure equal to the external. This vehicle can also be equipped with two arms, allowing the vehicle to perform object manipulation tasks through tele-operation.

Figure 1: GARBI underwater vehicle at sea.

The vehicle incorporates 4 thrusters: two for performing horizontal movements (yaw motion) and two for vertical movements (Z axis). Due to the distribution of weight, the vehicle is completely stable in pitch and roll. For this reason the vertical and horizontal movements are totally independent. The robot has a color camera fixed to its prow. A geometric calibration of this camera has been carried out to obtain its intrinsic parameters. The vehicle also includes several sensors which so far have not been used in this work, with the exception of a sonar, which indicates the distance from the vehicle to the bottom of the sea. The dimensions of GARBI are 1.3 meters in length, 0.9 meters in height and a width of 0.7 meters. The vehicle has a maximum speed of 3 knots and its weight reaches 150 Kg.

3 Mosaic-Based Positioning

The creation of the mosaic is accomplished in four stages: feature selection and matching, estimation of dominant motion, homography computation and mosaic blending. A more detailed scheme of the algorithm is shown in figure 2, and is explained below.

3.1 Feature selection and matching

The searching for feature correspondences is performed in a two-step approach. First, the zones of the image presenting high spatial gradient information are selected by means of a corner detector. Then, the textural parameters of these areas of the image are used as a matching vector to be correlated with the next image in the sequence. Textures significantly help in the location of features in the image and are specially indicated for the underwater environment, where lack of image features and lighting variations are the norm. The set of textural features used in our implementation has been chosen for its suitability in underwater imaging. Moreover, since our system is equipped with a color camera, the corner detection and texture extraction procedures are fulfilled not only on the intensity image, but also on the hue and saturation components. Since hue and saturation have the property of scale-invariance, that is, H(R,G,B) = H(αR,αG,αB) and S(R,G,B) = S(αR,αG,αB), they are more stable to variations on the intensity of the illuminant [12], being adequate for processing in underwater imaging.

A simple corner detector has been especially developed for this application. The main idea is to detect points with a high spatial gradient in the X and Y directions [13]. The corner detector is applied independently to the Hue, Saturation and Intensity images, keeping three lists of corners, one for every image component.

Once the corners of image $I^{(k)}$ have been obtained, the algorithm searches for the candidate matches in the next image $I^{(k+1)}$. The matching process is accomplished in the following way (see figure 3): For every point $\mathbf{m}_j^{(k)}$ in image $I^{(k)}$ a correlation is performed by convolving a small

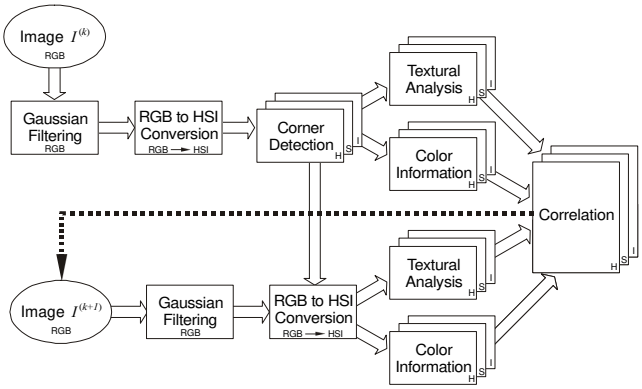


Figure 2: Scheme of the algorithm.

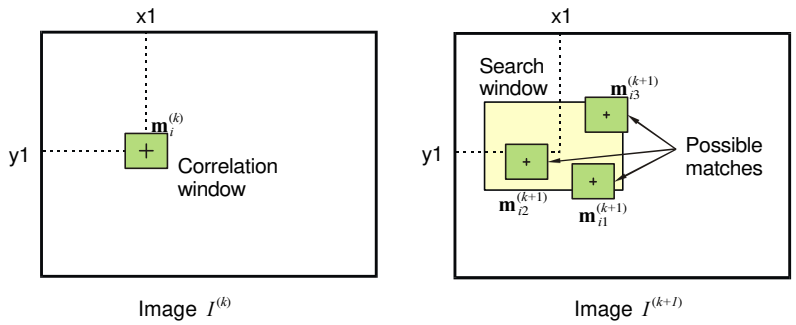


Figure 3: Typical situation where a corner point $\mathbf{m}_j^{(k)}$ has several possible matches in image $I^{(k+1)}$

window centered at $\mathbf{m}_j^{(k)}$ over a search window of image $I^{(k+1)}$. It should be noted that the correlation is performed on the same image component where the corner was detected. For instance, given a corner point $\mathbf{m}_j^{(k)}$ in the Saturation image $I_S^{(k)}$, a search for the best matches $\left\{\mathbf{m}_{j1}^{(k+1)}, \mathbf{m}_{j2}^{(k+1)}, ..., \mathbf{m}_{jq}^{(k+1)}\right\}$ is performed only in $I_S^{(k+1)}$. Only those matches that are quite similar to the original correlation window of $\mathbf{m}_j^{(k)}$ are taken into account. This similarity measurement is computed by means of the correlation score described in [14]. The threshold of the correlation score to be considered as a candidate match has been fixed to 0.85. Once the set of possible matches $\left\{\mathbf{m}_{j1}^{(k+1)}, \mathbf{m}_{j2}^{(k+1)}, ..., \mathbf{m}_{jq}^{(k+1)}\right\}$ has been obtained, the texture parameters of the patches centered at every matching point are computed (correlation windows on the right in figure 3).

Continuing with our example, for every possible match in the saturation image $I_S^{(k+1)}$, a vector of texture parameters is computed in the neighborhood of $\mathbf{m}_{j*}^{(k+1)}$. The texture parameters that have been used are: *Co-occurrence matrix* [15], *Energy filter* [16], and *Local Binary Patterns* [17]. We should take 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. In our application we chose 8 different angles for the co-occurrence matrix, taking only distances of 1 pixel, and 9 masks of the energy filter taking only a 3x3 neighborhood. From our experience, the use of larger neighborhoods provides little improvement at the expense of a higher computational cost. All the texture measurements are normalized between 0 and 1. A different vector is stored for the Hue, Saturation and Intensity images. Summarizing, the texture vector contains 18-positions, namely: 8 measurements of the co-occurrence matrix, 9 measures of energy, and 1 of the local binary patterns. These textures are only computed in the image component where the corner has been detected. If the corner $\mathbf{m}_j^{(k)}$ belongs to the image $I_S^{(k)}$, then the textures of every candidate matching $\mathbf{m}_{j*}^{(k+1)}$ are measured in the saturation image $I_S^{(k+1)}$. This texture vector is mapped onto a 18-dimentional space, where it is compared with the texture vector of the original point $\mathbf{m}_j^{(k)}$. The Euclidean distance is then computed, obtaining a texture similarity measure. After this process, a set of correspondences in image $I^{(k+1)}$ is obtained from every corner in image $I^{(k)}$, and every correspondence has two measures of similarity: correlation and texture. By averaging these two values, the reliability (r) of every match is obtained. Taking into account the reliability value, we have devised a method (the *disambiguate* algorithm) to correctly choose the right correspondence among the whole set of matches. In order to find the best correspondence for the jth corner $\mathbf{m}_j^{(k)}$, the disambiguate algorithm can be formulated as follows:

- Compute the centroid of the set of matches, weighing every coordinate depending on the reliability measure (r_{j*}):

$$\left(\bar{x}_j^{(k+1)}, \bar{y}_j^{(k+1)}\right) = \left(\frac{\sum_{i=1}^{q} x_{ji}^{(k+1)} \cdot r_{ji}^{(k+1)}}{q}, \frac{\sum_{i=1}^{q} y_{ji}^{(k+1)} \cdot r_{ji}^{(k+1)}}{q} \right) \quad (1)$$

- From the centroid $\left(\bar{x}_j^{(k+1)}, \bar{y}_j^{(k+1)}\right)$, fix a radius of size R, and eliminate those matches which fall outside of the circle defined by R, as shown in Figure 4.
- Select the match with the highest reliability as the correct correspondence.
- If no match appears inside the circle, then eliminate the corner.

Once this procedure has been accomplished, a set of pairs point-matching $\left\{ \left(\mathbf{m}_j^{(k)}, \mathbf{m}_j^{(k+1)}\right) \middle| j = 1, ..., p \right\}$ is obtained.

3.2 Selection of points describing the dominant motion

After the correspondences have been found, a set of displacement vectors relating the features of two images of the sequence is obtained. Every vector relates the coordinates of the same feature in both images. Although an accurate texture analysis is devoted to the matching procedure, some false matches (known as *outliers*) could still appear among the right correspondences. These false matches are mainly due to the presence of moving objects (algae or fishes) which violate the assumption of static scene, or even to the inherent system noise For this reason, a robust estimation method has to be applied. The Least Median of Squares (LMedS) algorithm aims at finding the affine transformation matrix **H** which minimizes the median of the squared errors. The matrix **H** describes the motion between two consecutive images. The minimization is performed by searching in the parameter space, and the error is defined by the distance of a point to the projection of its correspondence. Equation (2) expresses the non-singular linear transformation of the image plane into itself [18]:

$$\begin{bmatrix} x_j^{(k)} \\ y_j^{(k)} \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & -h_{21} & h_{31} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_j^{(k+1)} \\ y_j^{(k+1)} \\ 1 \end{bmatrix} \qquad (2)$$

where $\mathbf{m}_j^{(k)} = (x_j^{(k)}, y_j^{(k)}, 1)$ and $\mathbf{m}_j^{(k+1)} = (x_j^{(k+1)}, y_j^{(k+1)}, 1)$ denote a correspondence point in the present $I^{(k)}$ and next image $I^{(k+1)}$, respectively, expressed in homogeneous coordinates, $h_{11}, h_{12}, ..., h_{23}$ are 5 parameters that determine an affine transform; and $\cong$ indicates equality up to scale. Each point correspondence generates two equations, then at least 3 points are needed for the 5 unknowns.

Our implementation of the LMedS algorithm works as follows: given the problem of computing the homography matrix **H** from a set of data points, where 3 is the minimum number of data points which determine a solution, compute a candidate solution based on a randomly chosen 3-tuple from the data. Then, estimate the fit of this solution to all the data, defined as the median of the squared residuals. The median of the squared residuals is defined by:

$$M_{err} = \underset{j}{med}\left(d^2\left(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{H}\tilde{\mathbf{m}}_j^{(k+1)}\right)\right) + \left(d^2\left(\tilde{\mathbf{m}}_j^{(k+1)}, \mathbf{H}^{-1}\tilde{\mathbf{m}}_j^{(k)}\right)\right) \qquad (3)$$

where $\tilde{\mathbf{m}} = (\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)$ are the homogeneous coordinates of a 2D point $\mathbf{m}$ defined in the image plane, being $\mathbf{m} = (x_i, y_i) = (\mathrm{x}_1/\mathrm{x}_3, \mathrm{x}_2/\mathrm{x}_3)$ its corresponding Cartesian coordinates; and $d^2(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{H}\tilde{\mathbf{m}}_j^{(k+1)})$ is the square distance from a point $\tilde{\mathbf{m}}_j^{(k)}$, defined on image $I^{(k)}$, to the projection on the same image plane of its correspondence $\tilde{\mathbf{m}}_j^{(k+1)}$.

Once the best solution has been found, a minimal median is obtained. As from the median, the mean and the standard deviation can be computed (see [19] for details). Therefore, in our implementation, those points at a distance larger than the median are eliminated, and matrix **H** is recomputed with the remaining points, through a least squares criteria.

3.3 Mosaic construction

As soon as the best transformation **H** between two frames has been found, the two images are warped together composing the mosaic. The 2D motion of the camera is known in pixels from one image to the next, as an affine measure: rotation, translation and scaling. With the aid of an ultrasonic sensor, and the knowledge of the intrinsic parameters of the camera, 3D metric information about vehicle motion can be recovered. Although this metric information is relative to the sea bed, it can be very useful for navigation and mission planning. At the same time it introduces new knowledge on how the mosaic is evolving.

The experimental results on real images have been tested on the GARBI underwater vehicle [11], showing the effectiveness of the proposed method.

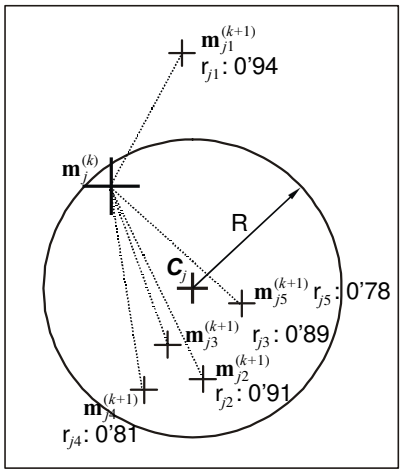
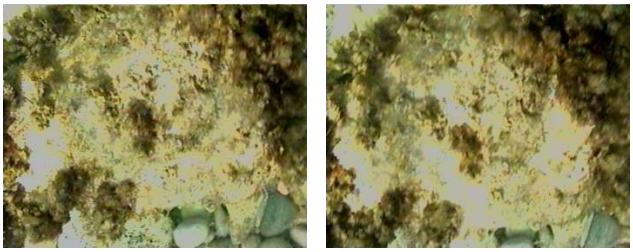


Figure 4: Given a corner point $\mathbf{m}_j^{(k)}$ every candidate correspondence $\mathbf{m}_{j^*}^{(k+1)}$ found in its neighborhood has an associated reliability $(r_{j^*}^{(k+1)})$. A weighed centroid (C_j) can be computed, defining a circle of radius *R*, where the right correspondence will be chosen.

4 Results

The sea bed mosaics presented in this paper were created from a set of images taken by the GARBI underwater vehicle. GARBI's camera pointed downwards to the ocean floor and the image capture rate was set to 4 images per second. The images were stored to disk and the construction of the mosaic was carried out offline. Figure 5



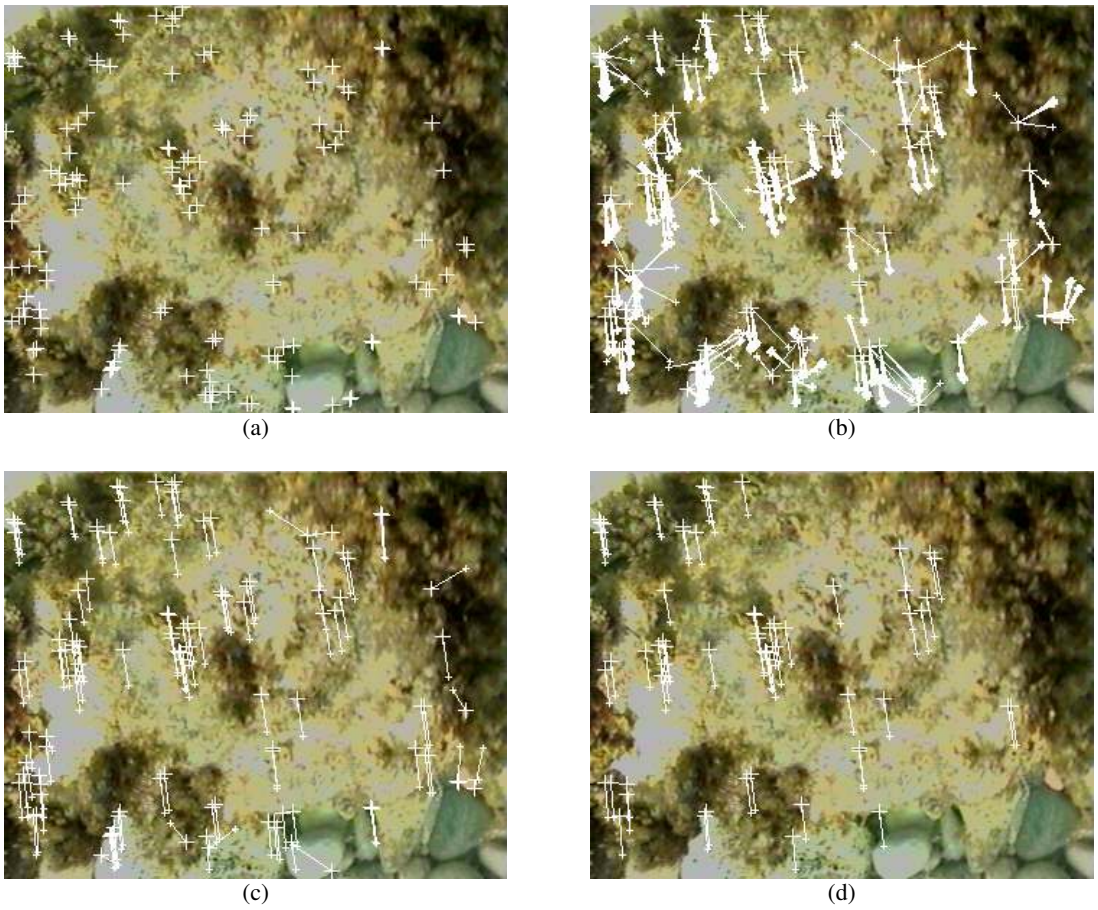
(a) (b)

Figure 5: Two consecutive images of the sequence: (a) image $I^{(k)}$; (b) image $I^{(k+1)}$.

shows the different phases accomplished by the mosaicking system. (a) and (b) are two consecutive images of the sequence. Notice the low-contrast of both images, and the difficulty of establishing matches directly from the intensity component. The hue, saturation and intensity components of the first image are computed to run the corner-detector algorithm. The resulting corners are shown in Fig. 6(a); while the neighborhood of these points is correlated against the next image of the sequence using again the components of intensity, hue, and saturation. Figure 6 (b) shows a num- ber of pairs point/matching after using the textural parameters. The larger crosses represent the position of every corner in the first image, while the possible matchings in the next image are represented with a small cross.

Figure 6(c) shows how the *disambiguate* algorithm, described in section 3.1, eliminates most of the incorrect matches, leaving only one match for every corner. This figure illustrates how the algorithm is not able to choose the correct matches in a small number of cases. However, it reduces the computational burden of the subsequent steps by providing a single match for every corner. The result of the LMedS algorithm applied to the remaining points is demonstrated in Figure 6 (d).

This algorithm is able to eliminate all the incorrect matches, although some of the correct ones are also deleted. Two example image mosaics are illustrated in Figure 7.



(a) (b)

(c) (d)

Figure 6: (a) corners detected in image $I^{(k)}$; number of corners: 138; (b) result of correlation (merging intensity, hue, saturation and textural parameters); number of pairs point/matching: 111; the corners are represented with the big cross, and the matchings are drawn with a small cross; (c) result of the *disambiguate* algorithm; (d) result of LMedS after *disambiguate*.

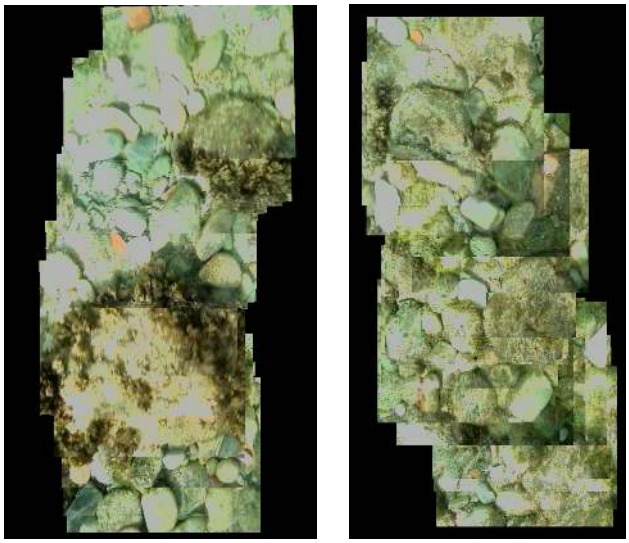


Figure 7: Sea bed mosaic examples.

5 Conclusions and Further Work

In this paper we have presented a mosaicking algorithm which is able to provide positional coordinates to an underwater vehicle. The viability of the system has been demonstrated with real world experiments. Textures, in addition to color, can highly improve the matching process, and robust estimation techniques further ameliorate the dominant motion estimation. The integration of a sonar measurement along with the visual information provides 3D position estimations of the submersible's motion.

Beyond this preliminary work, further research is needed to determine which are the texture parameters that better fit the matching process. Moreover, an effort is being made to ameliorate the throughput of the system by means of the development of special-purpose hardware for real-time motion estimation.

References

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

[2] N. Gracias and J. Santos-Victor "Automatic mosaic creation of the ocean floor", in *Proc. of the OCEANS Conf.*, vol. 1, pp. 257–262, 1998.

[3] R. Haywood, "Acquisition of a micro scale photographic survey using an autonomous submersible", in *Proc. of the OCEANS Conf.*, vol. 5, pp. 1423–1426, 1986

[4] R.L. Marks, S.M. Rock, and M.J. Lee, "Real-time video mosaicking of the Ocean floor", *IEEE Journal of Oceanic Engineering*, vol. 20, no.3, pp. 229–241, 1995.

[5] S. D. Fleischer, H. H. Wang, S. M. Rock, and M. J. Lee, "Video Mosaicking Along Arbitrary Vehicle Paths", in *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pp. 293–299, 1996.

[6] S. Negahdaripour and J. Fox, "Underwater optical station-keeping: improved methods", *Journal of Robotic Systems*, vol. 8, no. 3, pp. 319–338, 1991.

[7] S. Negahdaripour, C. H. Yuh, "A generalized brightness change model for computing optical flow", in *Proc. of the IEEE Int. Conf. on Computer Vision*, 1993.

[8] S. Negahdaripour and L. Jin, "Direct recovery of motion and range from images of scenes with time-varying illumination", in *Proc. of the Int. Symp. of Computer Vision*, 1995.

[9] S. Negahdaripour, X. Xu, L. Jin, "Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 370–382, 1999.

[10] J. Amat, J. Batlle, A. Casals and J. Forest, "GARBI: A Low Cost ROV, Constraints and Solutions", in *Proc. of the 6th IAPR Underwater Robotics*, 1996.

[11] Amat, J. Batlle, J. Montferrer, A. Salvi, J. and Ridao, P., "Capabilities of GARBI - A Low cost underwater Vehicle", in *Proc. IEEE/RSJ Int. Conf. on Robots and Systems*, 1998.

[12] J. Regincós and J. Batlle, "A System to Reduce the Effect of CCDs Saturation", in *Proc. IEEE Int. Conf. on Image Processing*, pp. 1001-1004, 1996.

[13] L. Kitchen, A. Rosenfeld, "Gray-Level corner detection", *Pattern Recognition Letters*, vol. 1, no. 2,pp. 95-102, 1982.

[14] Z. Zhang, R Deriche,. O. Faugeras, Q.T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", INRIA RR-2273, 1994.

[15] 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, November, 1973.

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

[17] 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.

[18] R. Szeliski, "Image mosaicing for tele-reality applications", in *Proc. of the IEEE Workshop on Applications of Computer Vision*, pp. 44–53, 1994.

[19] P. Rousseeuw and A. Leroy, "Robust Regression and Outlier Detection", John Wiley & Sons, New York, 1987.