Coastal Monitoring and Feature Estimation with Small Format Cameras: Application to the Shoreline of Monte Hermoso, Argentina

Natalia Revollo[1,2], Claudio Delrieux[3] , Gerardo Perillo[1,4] and Marina Cipolletti[1,3]

[1] Instituto Argentino de Oceanografía, *CONICET*
Camino a la Carrindaga km 7 Complejo CCT Edificio E1 B8000FWB,
Bahía Blanca Argentina iado@criba.edu.ar
[2] Facultad de Ingeniería, Universidad Nacional de Jujuy, *UNJU*
Gorriti 237 San Salvador de Jujuy
Jujuy Argentina
[3] Instituto de Investigaciones en Ingeniería Eléctrica-Dpto. de Ing. Eléctrica y de Computadoras, UNS-CONICET
Avenida Alem 1253 B8000CPB
Bahía Blanca Argentina iiieuns@uns.edu.ar
[4] Departamento de Geología, UNS
San Juan 670, Bahía Blanca Argentina geologia@criba.edu.ar

Abstract. Image and video processing of natural phenomena is one of the preferred non-invasive monitoring techniques for environmental studies that is, however, limited through the high cost of the required equipment and the limited access and precision of the processing algorithms. In this work we propose a low cost methodology for environmental studies using unexpensive off-the-shelf hardware and simple yet powerful processing algorithms. The images are taken using small format RGB cameras and processed in standard laptop equipments using open source libraries and processing algorithms specifically developed in general purpose programming languages. We applied this methodology to the coastal monitoring the shoreline of Monte Hermoso, Argentina, aimed at establishing accurate measurements of specific coastal features, for instance the coastal length. The experimental results show that our proposed unsupervised processing algorithm obtains results with a very high level of accuracy.

Keywords: Image and video processing. Environmental monitoring. Reprojection, segmentation, and feature extraction algorithms.

1 Introduction

Image and video processing of natural phenomena, together with remote sensing, are among the preferred non-invasive monitoring techniques for obtaining qualitative and quantitative information that may be used for environmental, economic, and social decision making, and establishing policies. Remote sensing imagery (*i.e.*, satellite or airborn), however, is not versatile, often not precise enough, and may have very high operational costs. Monitoring using computer vision, on the other hand, may overcome these limitations when two conditions are met: if they are based on affordable, easily replaceable off-the-shelf equipment, and if the processing algorithms are reliable.

The long term goal of our study is to establish an accurate model of the sea-wave-land interaction in the shoreline of Monte Hermoso, Argentina. There are other worldwide projects that study the shoreline dynamics, for instance ARGUS (Holman and Stanley, 2007), INDIA (Morris et al., 2001), HORS (Takewaka et al., 2002), CAM-ERA project, KOSTA project and HORUS project (Andrés Osorio et al., 2007). These research projects present results of shoreline dynamics, but unfortunately most of the image processing algorithms that they implement are not well documented.

This work presents itself as the first results of beach shoreline dynamics research employing low cost small format cameras and image processing in Argentina. Determination of several coastal features and their dynamics (i.e., the perimeter value of the polygonal that represents the shoreline) allows the study of several geomorphologic processes, and the elaboration of a geophysical predictive model that may serve in several decision making instances.

Images are acquired from a static vintage point in a building. The processing pipeline requires very little tuning to be fully unsupervised. First, images are rectified to a zenithal plane, and linearly georeferenced using four GPS reference points. After this step we can guarantee that the raster scale is even enough for a further vectorization and quantification process.

Then the image is binarized with respect to proximity to a prototype in color space. The purpose of this binarization is to classify every pixel in the raster as either sea or land. The binarized image is then processed with a set of morphology filters to even out noise or spurious misclassified pixels.

Finally, we applied the usual linear length extraction methods to the border between the classified areas in the raster. We argue that these methods, though popular, incur in a significant, systematic error in excess, which is also resolution independent. For this reason, we applied also a super-resolution variant of the Marching Squares algorithm to measure the linear length of the coastal shoreline. These length estimations were tested against the length of a polyline drawn by hand that visually fits the shoreline tightly (and therefore considered here as the gold standard). Results show that our unsupervised method is able to provide a very precise and accurate estimation of the coastal length.

2 Methodology

2.1 Image Rectification Stage

Data Acquisition. The digital images used in the rectification process were taken with unexpensive small format digital RGB camera from the beach of Monte Hermoso in Buenos Aires Province, Argentina. The camera was located in a fixed position at a height of 30 m on top of a building, having a panoramic view of the beach. The obtained images have a 1293x1142 resolution.

Camera Model. In oblique images, the scale of the raster varies along the position, making difficult the measurement of geometric variables under study. For this reason, a camera model is used to relate points in the image to their corresponding geographic coordinates. A two-dimensional projective transformation was used to transform the oblique projection plane to a zenithal plane (Lerma, 2002). This situation is characteristic in the rectification of photographic images (see Fig. 1).

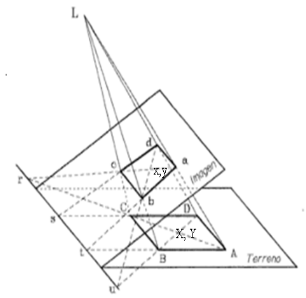

Fig. 1. Two-dimensional projective transformation of two nonparallel planes. The X and Y coordinates represent the corresponding geographic coordinates of a point (x, y) that is projected in the oblique plane of a digital image.

The expressions of the two-dimensional projective transformation (**1**) express the projection between two nonparallel planes from eight parameters.

$$[A] = \begin{vmatrix} a_1 b_1 c_1 \\ a_2 b_2 c_3 \\ a_3 b_3 c_3 \end{vmatrix} \neq 0 \qquad \begin{gathered} x = \frac{a_1 X + b_1 Y + c_1}{a_3 X + b_3 Y + 1} \\ y = \frac{a_2 X + b_2 Y + c_2}{a_3 X + b_3 Y + 1} \\ \begin{bmatrix} x \\ y \end{bmatrix} = [A] \begin{bmatrix} X \\ Y \end{bmatrix} \end{gathered} \tag{1}$$

Image Rectification. An oblique image rectification using two-dimensional projective transformation needs four homologous points. In this task four control points in the oblique image and four real land points were needed. These points allow calculating the projection matrix. Four control points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) were determined in the oblique image. These points limit the geographic area of interest (shown in Fig. 2). The geographic coordinates of the control points in the image were obtained using a Global Positioning System. For each control point the coordinates (x, y) in pixel on the image was determined using a GIS software, whose values are shown in Table 1.

Fig. 2. Control Points $((x_1, y_1)$, (x_2, y_2), (x_3, y_3), $(x_4, y_4))$ whose geographic coordinates were obtained with a Global Positioning System and are related with the pixels coordinates in the image.

Table 1. Pixel Coordinates of the control points in the image.

Points	*x*	*y*
Point1 (x_1, y_1)	367	448
Point2 (x_2, y_2)	507	407
Point3 (x_3, y_3)	1043	610
Point4 (x_4, y_4)	637	866

The GPS-established coordinates are shown in Table 2, and were positioned on an information layer in Google Earth (Fig. 3). The geographic distance between these points was obtained, and the pixel coordinates in the image of the projected real points were determined are shown in Table 2. The pixel values of the four control points on the oblique image and the four geographic points allow to calculate the projection matrix and then the projective transformation for the rectification.

Fig. 3. GPS-established control points taken in the image. These geographic coordinates determine the parameter of the projection matrix.

Table 2. Geographic coordinates of the control points in the image.

Points	*Latitude*	*Longitude*
Point1 (X_1, Y_1)	38º 59’ 21,27’’	61º 16’ 36,27’’
Point2 (X_2, Y_2)	38º 59’ 28,65’’	61º 16’ 37,39’’
Point3 (X_3, Y_3)	38º 59’ 27,40’’	61º 17’ 19,82’’
Point4 (X_4, Y_4)	38º 59’ 21,27’’	61º 17’ 18,88’’

Image rectification was implemented in a standard general-purpose programming language, using the Free Vision Library OpenCV, that implements a variety of tools and functions used in the image processing and computer vision in real time. The rectification procedure takes the oblique image and the matrix parameters, and produces as output the rectified image that will be used for later processing (see Fig. 4).

2.2 Image Segmentation Stage

The significant part of the rectified image including the shoreline area was clipped into a new image (see Fig. 5). In this image, further processing will be applied to extract the shoreline, and to compute an adequate estimation of its length.

Image Classification. The purpose of this processing is to determine the likelihood of a pixel being either sea or land. A prototype of sand color was determined (Table 3), and the distance in color space to this prototype was established for every pixel in the image. The resulting pseudocolored image can be seen in Fig. 6. After this classification step, the likelihood of a pixel as being sea is proportional to the distance to the sand color prototype (and therefore to the grey level in Fig. 6). This secondary, “distance” image can be binarized against a threshold, producing the desired classification.

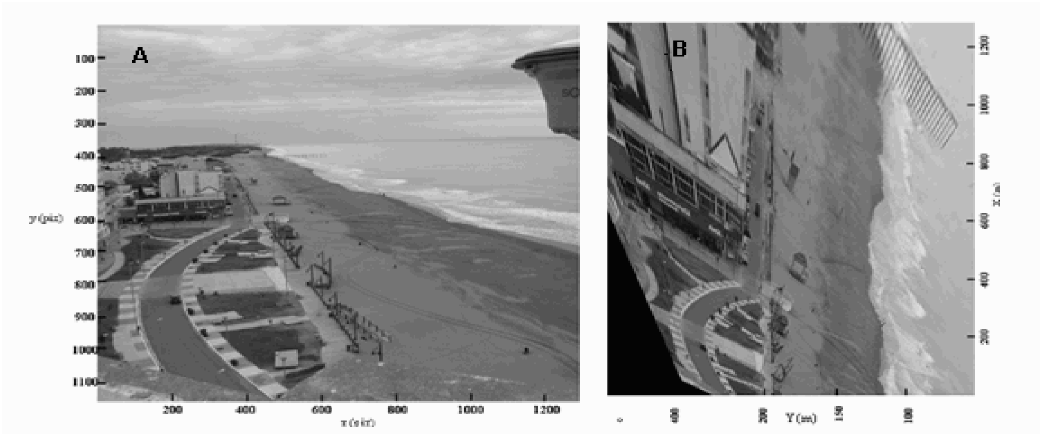

Fig. 4. a) Oblique image of Monte Hermoso where known geographic coordinates, and b) previous image after rectification (shown offscale).

Table 3. Prototype pixel Coordinates and RGB values.

Prototype Pixel(Position 93,839)	*value*
Red Component	99
Green Component	93
Blue Component	85

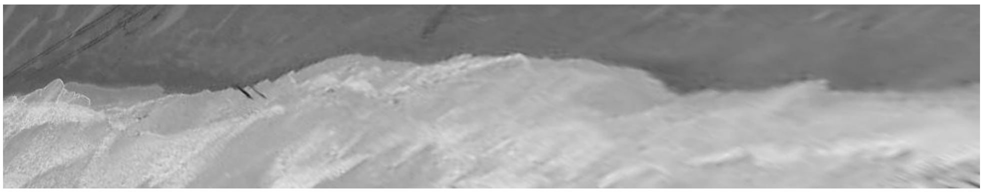

Fig. 5. A clip of the rectified image containing the area of interest.

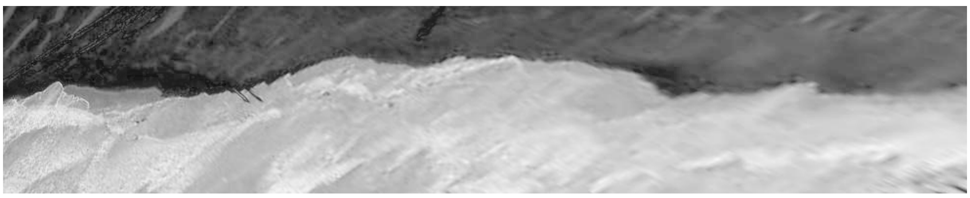

Fig. 6. Pseudocolor image where the gray level is related to the distance in color space of the pixel color to a prototype.

Usual threshold setting techniques are based on a histogram analysis of the relative frequency distribution of distances. The criterion of minimizing together the amount of false negatives and positives gives rise to the minimum distance classifier, and the criterion of minimizing the conditional risk of misclassification gives rise to the Bayesian classifier.

In our images, however, neither of these criteria meets fully the desired result of a cleanly segmented shoreline. For this reason, we set an empirically tuned threshold value that produced the best results in the following processing steps in the pipeline (see Fig. 7), *i.e.,* removing the undesired misclassified areas that are visible in the uppermost right part of the image (false positives), and the ones that are also seen in the lowermost left part (false negatives).

Fig. 7. Images binarized with different threshold values. Empirically, the middle one performs best in the further processing steps.

Noise and misclassified areas removal. Noise and misclassified areas in the binarized image obtained in the previous step (Fig. 7(b)) was removed by means of morphological filtering (dilation and erosion). The resulting image (Fig. 8) is clean enough for the border extraction and measurement step, and therefore is taken as the correctly classified image.

Fig. 8. Correctly classified image after noise and misclassified pixel removal.

Border Extraction. The extraction of the polygonal that represents the shoreline was computed using several methods. The two most popular ones (implemented in almost all GIS software) are the outermost edge detection method, and the chain code method. The outermost edge detection method regards every edge between pixels classified in the interior area and the exterior area of the classified image as part of the border between these two areas. The chain code method surrounds the classified area with a polyline, every line of which may be oriented in any of the eight principal angles.

It is easy to see that, though often used, these methods incur in a rather high systematic error in excess, which even worse is resolution independent. For this reason, we devised a super-resolution method based on the marching squares algorithm, but that takes into account the classification distance of the pixels prior to binarization, therefore being able to do a much finer segmentation of the border (see Fig. 9).

Supervised Measurement using a (GIS) Software. Also, a supervised measurement of the shoreline was made with a GIS software tool. The software employed was GvSig, this is a free software developed by the Generalitat Valenciana. The rectified image was loaded in the system and then a new vectorial layer of the shoreline was digitalized by hand. The length of the resulting polyline is then taken as the gold standard against which the accuracy and precision of our unsupervised methods are tested.

3 Results and Discussion

The lengths of the shoreline as measured with all the methods implemented in this paper are shown in Table 4. As predicted, both the outermost borders, and the chain code methods commit an unacceptable error in excess. The super-resolution marching squares algorithm, however, is only about 2.5% above the supervised measurement. This may be attributed to the fact that the tendency in user assisted vectorization is always to obliterate tiny details, and therefore it is most likely that the actual length of the shoreline is slightly above the reported by the supervised method.

Table 4. Length obtained using different algorithms of measurement

Algorithm	*Length*
Manual segmentation	1193
Outermost borders	1558
Chain Code	1322
Super-resolution Marching Squares	1225

The behaviour of the four methods can be appreciated in detail in Fig. 9, where we superimposed the four different versions of the border over a close-up of a clip of the rectified image containing part of the shoreline.

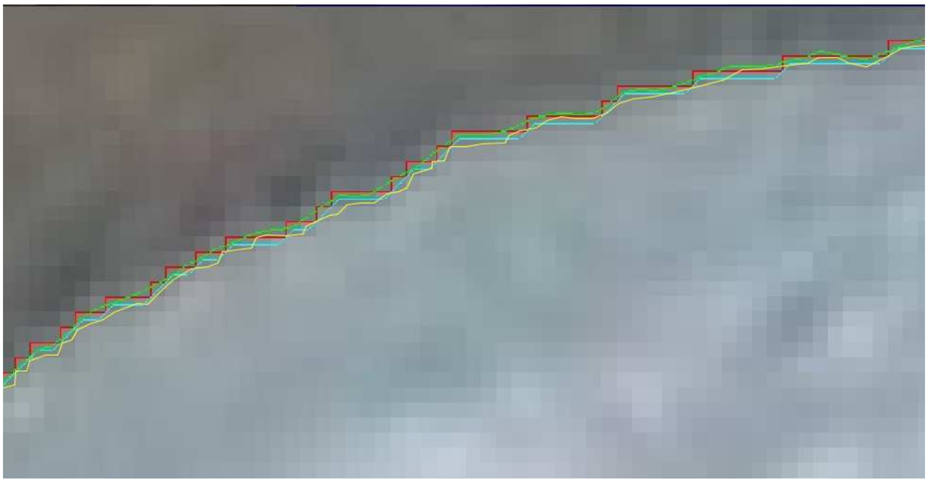

Fig. 9. The borders obtained with the four methods superimposed over a close-up of the original image: outermost borders (red), chain code (green), super-resolution marching squares (cyan), and supervised (yellow).

4 Conclusion and further work

Image and video processing of natural phenomena using unexpensive equipment posses a significant challenge in environmental studies. In this work we implemented a coastal monitoring and feature estimation system using unexpensive, small format RGB cameras. The methodology was applied to measure shoreline lengths in the area of Monte Hermoso, Argentina.

Our processing framework acquires the images from a static vintage point in a building. The images are then rectified using a reprojection transformation to a zenithal plane, and then they are georeferenced using the projection of four locations whose GPS coordinates are known. Then the pixels in the rectified image are classified in the RGB color space with respect to proximity to a prototype. Finally, the length of the border is estimated using three different algorithms.

The experimental results were tested against a supervised estimation, and show that our proposed unsupervised processing algorithm obtains highly accurate results, arguably as good as the supervised ones.

The processing pipeline requires only a small set of externally fixed parameters (the position and projection of four geographic coordinates, the color prototype against which the classification is performed, and the empirically selected threshold value for binarization). These parameters are robust enough along whole sequences of images, and therefore the entire process is amenable for processing video sequences lasting several minutes.

Several research lines are opened by these results. Among them the most important one is to develop a portable, low-cost, easy to operate monitoring station that is able to acquire and process video sequences along several hours. For this purpose, some strategies for having adaptive tuning of the processing parameters will be required, specifically for making the classification independent to illumination and weather conditions. Also important is a drastic optimization of the execution time of the algorithms, in order to have an adequate frame per second processing rate. Finally, the implementation of these algorithms in an embedded system that might be able to connect to a repository server through GSM connectivity, will enable outstanding real-time environmental monitoring capabilities.

5 References

C. González Rafael and Richard E. Woods, "Tratamiento Digital de Imágenes", Ed. Addison-Wesley/Díaz de Santos, 1988.

M. Browne, M. Blumenstein, M. Tomlinson, "An Intelligent System for Remote Monitoring and Prediction of Beach Safety Proceeding Artificial Intelligence and Applications", R. Lane 453, 2005.

David Chu, Lucian Popa, Arsalan Tavakoli, Joseph Hellerstein, Philip Levis, Scott Shenker, and Ion Stoica. "The Design and Implementation of a Declarative Sensor Network System." Proceedings of the Fifth ACM Conference on Embedded Networked Sensor Systems (SenSys), 2007.

José Luis Lerma García: "Fotogrametría Moderna: Analítica y Digital". Editorial Universidad Politécnica de Valencia, 2002.

Laura Uunk, "Automated collection of intertidal beach bathymetries from Argus video images" MSc Thesis, 2006.

Andrés F. Osorio, Juan C. Pérez, Cristian Ortiz and Raúl Medina, "Técnicas basadas en imagines de video para cuantificar variables ambientales en zonas costeras", Avances en recursos hidráulicos, Numero 16, 2007.

Intel Corporation "Open Source Computer Vision Library" (OpenCV), http://www.intel.com/technology/computing/opencv/

Josefa García León, Aurora Cuartero Sáez, "Comparación de los procesos de rectificación y ortoproyección mediante fotogrametría terrestre Digital" Universidad de Extremadura España.

R. Holman, J. Stanley, and T. Ozkan-Haller, "Applying Video Sensor Networks to Nearshore Environment Monitoring", Pervasive Computing 2:14-21,2003.