Journal of Electronic Imaging

JElectronicImaging.org

Heuristic method based on voting for extrinsic orientation through image epipolarization

Santiago Martín
José Luis Lerma
Hodei Uzkeda

SPIE. IS&T imaging.org

Santiago Martín, José Luis Lerma, Hodei Uzkeda, "Heuristic method based on voting for extrinsic orientation through image epipolarization," *J. Electron. Imaging* **26**(6), 063020 (2017), doi: 10.1117/1.JEI.26.6.063020.

Heuristic method based on voting for extrinsic orientation through image epipolarization

Santiago Martín,[a] José Luis Lerma,[b] and Hodei Uzkeda[c,*]
[a]Universidad de Oviedo, Departamento de Construcción e Ingeniería de Fabricación, Campus de Viesques, Gijón, Spain
[b]Universitat Politècnica de Valéncia, Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Valencia, Spain
[c]Universidad de Oviedo, Departamento de Geología, Oviedo, Spain

Abstract. Traditionally, the stereo-pair rectification, also known as epipolarization problem, (i.e., the projection of both images onto a common image plane) is solved once both intrinsic (interior) and extrinsic (exterior) orientation parameters are known. A heuristic method is proposed to solve both the extrinsic orientation problem and the epipolarization problem in just one single step. The algorithm uses the main property of a coplanar stereopair as fitness criteria: null vertical parallax between corresponding points to achieve the best stereopair. Using an iterative approach, each pair of corresponding points will vote for a rotation axis that may reduce vertical parallax. The votes will be weighted, the rotation applied, and an iteration will be carried out, until the vertical parallax residual error is below a threshold. The algorithm performance and accuracy are checked using both simulated and real case examples. In addition, its results are compared with those obtained using a traditional nonlinear least-squares adjustment based on the coplanarity condition. The heuristic methodology is robust, fast, and yields optimal results. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.6.063020]

Keywords: epipolarization; epipolar geometry; extrinsic orientation; image rectification; relative orientation.
Paper 170127 received Feb. 17, 2017; accepted for publication Nov. 7, 2017; published online Nov. 28, 2017.

1 Introduction

In a classic stereo-photogrammetry project, the possibility of reconstructing the relative pose (i.e., extrinsic orientation parameters, position, and orientation) of two calibrated cameras, as well as the locations of a set of points in space, from the projection of those points onto both images is well known. The epipolar geometry of two views provides the framework to do that.[1–4] It is based on just one main geometric property: the coplanarity condition satisfied by the projection centers, object point, and image points. Given at least five points, this constraint is enough to solve the camera pose.[5–7]

The gold standard for estimation of the extrinsic orientation parameters is to perform a bundle adjustment with a high number of images. This allows one to overcome issues derived from lack of redundancy which may mask gross errors.[8] For stereopairs, the relative orientation is used to determine the extrinsic orientation parameters, following the classic nonlinear least-squares adjustment either with the coplanarity condition or the collinearity condition.[5–7] As it is well-known, least-squares estimation leads to satisfactory convergence when approximate camera pose estimates are close to the final solution, providing precise results. On the other hand, if the initial estimates are not close enough to the final solution, this method is sensitive to local (false) minima, what may lead either to a wrong solution or to a nonconvergence of the estimated extrinsic orientation parameters. Therefore, direct methods are indicated to carry out the initial estimates.[9]

The pose information between a pair of calibrated camera images is algebraically captured by an essential matrix. The most well-known direct methods are named based on the number of points that they require.[10] For the task of obtaining the relative orientation of two views with calibrated intrinsic parameters, which entails estimating the relative rotation and the translation direction between the two views, the minimal number of point correspondences required in general circumstances is five.[2,11] Nevertheless, for many special motions, the problem may be solved with even fewer points.[2] It can be shown that there are up to 10 solutions, though the solutions are not obtainable in closed form. Other algorithms use six points[12] or even more. Seven points lead to up to three solutions, whereas eight points provide a linear equation for a unique solution.

Once the camera pose problem is solved, assuming that the internal camera parameters are known, epipolarization is possible. The epipolarization of a convergent stereopair consists on reprojecting both images onto a common image plane. Stereo-pair epipolarization is used in computer stereo vision to simplify the problem of finding matching points between images, because search is done along the horizontal lines of the rectified images. Following the epipolar lines, a depth map (which contains information regarding the distance between any pixel in the two-dimensional original image and the viewpoint) can be computed faster.

Although there are exceptions, most stereo-pair epipolarization techniques assume that the stereopair is calibrated, i.e., both intrinsic (interior) and extrinsic (exterior) orientation parameters are known.[13] In this paper, a heuristic method based on voting is proposed to solve both the extrinsic orientation and the epipolarization in just one single step. The algorithm assumes that the camera is calibrated (interior parameters known), but extrinsic parameters are

*Address all correspondence to: Hodei Uzkeda, E-mail: hodei@geol.uniovi.es

1017-9909/2017/$25.00 © 2017 SPIE and IS&T

not available. It uses an iterative approach, where each pair of corresponding points will vote for a rotation axis that may reduce vertical parallax.

Some researchers used previously optical flow algorithms for camera motion estimation, mainly working with mobile robots. Using a ground-facing monocular camera, the movement of a robot on a flat surface can be estimated from the pixels displacement between frames.[14–16] The video created with a front-facing camera in a sewer inspection system can be analyzed through optical flow vectors to recover information about travelling distance, position inside the sewer, and direction of motion.[17] The method proposed is founded in an optical flow. Nevertheless, as far as we know, the method proposed in this paper is completely new. The rest of the paper is organized as follows. Section 2 presents the heuristic approach based on voting, starting with an overview, analysis of rotation effects on y image coordinates, and finally the conceptual algorithm description. Section 3 deals with two case studies, one simulated and another real. Section 4 discusses the results. Finally, Sec. 5 draws some conclusions.

2 Method

2.1 *Overview*

The algorithm uses the main property of a coplanar stereopair as fitness criteria: null vertical parallax between corresponding points to achieve the best stereopair. Using an iterative approach, each pair of corresponding points will vote for a rotation axis that would reduce vertical parallax (fitness criteria). The votes will be weighted, the rotation applied, and an iteration will be carried out, until the vertical parallax residual error is below a threshold. The vote of each corresponding point depends on the area within the image where it lies.

Up to a point, the algorithm resembles an optical flow approach. The effect of a rotation in the y-coordinate of a point depends on: the rotation axis (X-, Y- and Z-axes), the focal length, and the initial image coordinates (x, y) of the point (i.e., the area the point belongs to). Some combinations of rotation and area increase the y-coordinate; others reduce it; and others do not alter its value. The method proposed is founded in an optical flow analysis of these rotations.

It is accepted that once both intrinsic and extrinsic orientation parameters are known, it is possible to solve the rectification problem. Once a stereopair is coplanar, the correspondence problem (i.e., finding a corresponding point viewed by one camera in the image of the other camera) is simplified to one dimension (Fig. 1). Two corresponding points should lie in a common horizontal line, parallel to the vector of translation between the cameras. There is no vertical parallax (i.e., there is no y-coordinate difference) between corresponding points in an epipolarized pair.

Finding the coplanar arrangement of two images can be solved finding two rotation matrices, one for each photo. These two matrices may be combined, as other methods do, into only one with the relative pose change. The translation vector (baseline) is parallel to the initial horizontal vector, common to both cameras before any rotation (i.e., vector X). There are infinite coplanar solutions, due to the five degrees of freedom related to either rotations or rotations/translations.

If one of the two images is rotated and epipolarized toward an image plane, the vertical y differences between corresponding points will change. The effect of a rotation in the y-coordinate of a point depends on the rotation axis, the focal length, and the initial image coordinates (x, y) of the point. If we analyze the effect of rotations around X-, Y-, and Z-axes, we will discover that they have different meanings on the y-coordinate of any point, according to the area the point belongs to. Some rotations on specific areas will increase the y-coordinate, whereas others will reduce it and the rest will not alter its value.

This fact is the core idea of the heuristic method based on voting proposed herein. The voting technique has been widely and successfully employed in photogrammetry and related fields of study, see for instance.[18–22] In this case, pairs of points give their votes considering the effect that the proposed rotation has over their vertical parallax. Using an iterative approach, each pair of corresponding points will suggest the rotation axis and direction that would reduce their y difference, just taking into account the area the points belongs to. The votes will be weighted, the rotation applied, and an iteration will be carried on until the rectification process achieves an acceptable result, i.e., the vertical parallaxes are below a defined threshold.

2.2 *Rotation Effects on y Image Coordinates*

It has been previously stated that if we analyze the effect of rotations around X-, Y- and Z-axes, we will discover that they have different effects on the y-coordinate of any point, according to the area the point belongs to. Given an object point $P(X, Y, Z)$ referred to the camera system

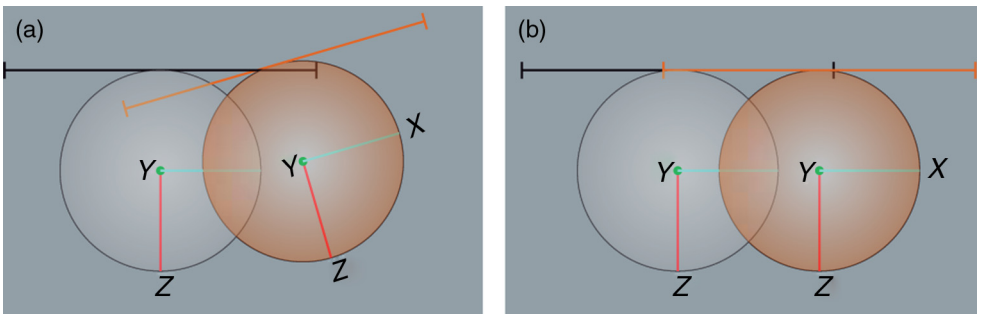

Fig. 1 Top view of the rectification problem: (a) slightly convergent stereopair and (b) epipolarized (normal) stereopair. Spheres represent the focal length of each camera. Each image plane can rotate around its sphere as long as it remains tangent.

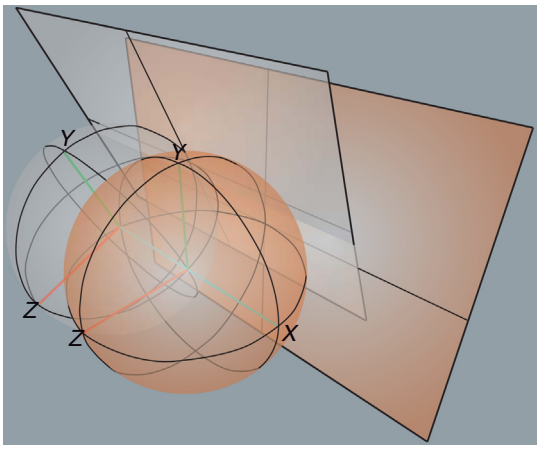

Fig. 2 Rotation around X-axis on the left image. The sphere represents the focal length of the camera. The image planes, rectangles in this figure, are tangent to the spheres.

(Fig. 2), the y image coordinate of the projection point $p(x, y)$, in the same reference system, can be calculated as Eq. (1) using the simplified formulation

$$y = -f\frac{Y}{Z}, \tag{1}$$

where f represents the focal length of the camera.

Rotating the image to a plane is equivalent to rotating the point P the same angle but in the other direction. In fact, a rotation of an angle A around the X-axis yields to the (X', Y', Z') coordinates of P calculated in

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2}$$

Using Eq. (2), the y'-coordinate of the projected point after rotation is calculated from the rotation angle A, the focal length f, and the coordinates (x, y) of the projected point before rotation using

$$\begin{aligned} y' &= -f\frac{Y'}{Z'} = -f\frac{Y\cos A - Z\sin A}{Y\sin A + Z\cos A} \\ &= -f\frac{y\cos A + f\sin A}{y\sin A - f\cos A}. \end{aligned} \tag{3}$$

In a similar way, the y'-coordinate of the projected point calculated after a rotation around the Y-axis is presented in Eq. (4) and around the Z-axis in Eq. (5)

$$y' = \frac{yf}{x\sin A + f\cos A}, \tag{4}$$

$$y' = x\sin A + y\cos A. \tag{5}$$

There are important differences between the three results, which are obvious from the equations obtained. While the result of the rotation around the Y-axis is affected by f, x, y, and A; the rotation around the X-axis is not affected by x; and the rotation around Z-axis is independent of f.

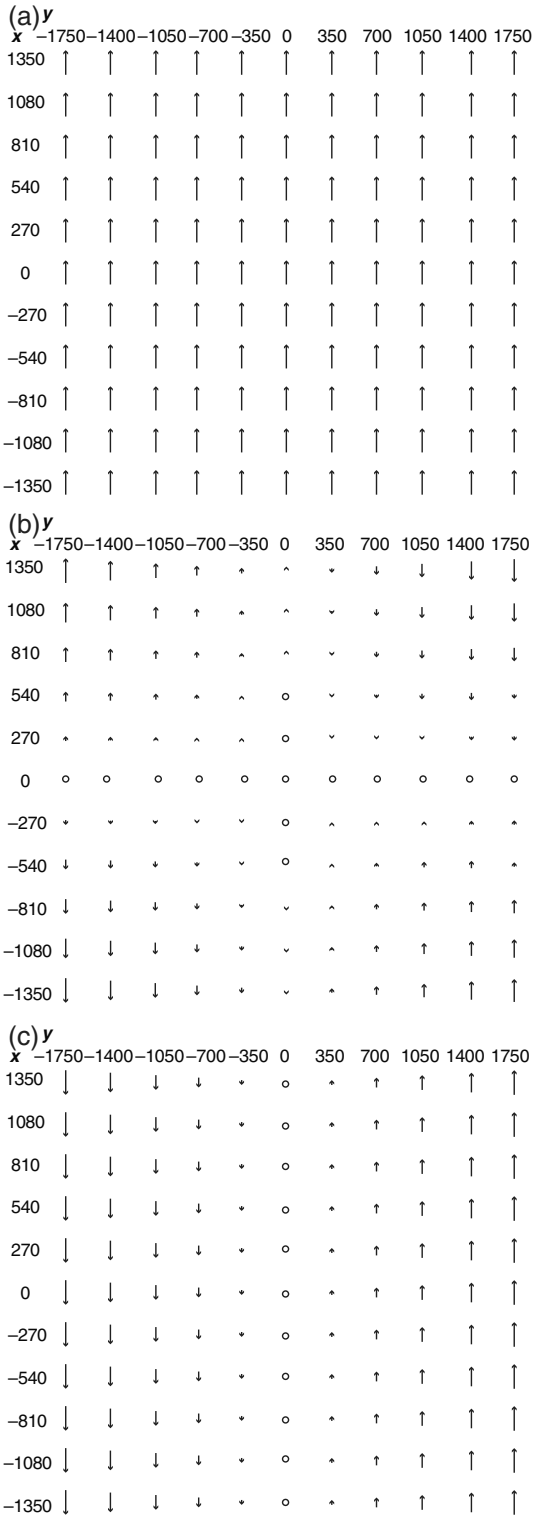

Fig. 3 Representation of pixels y' flow as a result of different rotations: (a) 0.03 deg around X-axix; (b) 0.70 deg around Y-axis; and (c) 0.13 deg around Z-axis. Vectors' length represents the amount of change (ranging from 0 to 4 pixels). Note that xyz-coordinate system origin is at the center of the image.

Taking a camera with a focal length of 7500 and image width, height of (3500, 2700) (all units expressed in pixels), Fig. 3(a) draws the effects on the y'-coordinates after a rotation around X-axis of 0.03 deg. Each y'-coordinate increases around 4 pixels, the change is almost irrespective to the

original (x, y) coordinates, i.e., original position of the point within the image.

Figure 3(b) shows the effects on y'-coordinate after a rotation around Y-axis of 0.70 deg (much bigger). Now, some y'-coordinates are modified up to 4 pixels, whereas others remain basically the same. The results vary highly depending on the original position (x, y), i.e., the quadrant the point occupied before the rotation.

Finally, Fig. 3(c) shows the effects on y'-coordinate after a rotation around Z-axis of 0.13 deg (intermediate value). As with the rotation around Y-axis, the y'-coordinates undergo a change ranging from 0 to 4 pixels, and the variation depends highly on the quadrant of (x, y). Next to both horizontal and vertical axes variations are negligible. These different behaviors allow separation of four major areas (quadrants) and other minor ones around both axes.

2.3 *Conceptual Algorithm Description*

The method proposed is an iterative algorithm (Fig. 4), where a set of corresponding points vote supporting a certain rotation around X-, Y-, or Z-axes for any of the photographs. Their votes are based on trying to minimize their vertical parallaxes and take into account the quadrant each point belongs to. The result will be two rotation matrices, one for the left image and another for the right one, which reduce the vertical parallaxes between corresponding points to a minimum. These matrices can be combined in a single rotation matrix between the two cameras.

A set of corresponding points, plus the intrinsic orientation parameters, including the lens distortion parameters, are given to the solver. The first step is "voting," as expressed in Fig. 4. Voting implies several steps. First, for each corresponding point, i.e., $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, the vertical parallax is calculated as $(y_2 - y_1)$. A positive vertical parallax means that the vote counting for P_1 will try to increase y_1 (and P_2 vote counting will try to decrease y_2). If the absolute vertical parallax is lower than a certain threshold value (called "MaxYError"), no difference is considered.

The next step of the algorithm is to classify both P_1 and P_2 in the seven standard areas per image (Fig. 5). Four quadrants (main areas) are initially considered per image. In addition, three minor areas are defined around both horizontal and vertical axes. These latter areas have a width and height of just a few pixels around the axes. The parameter "axial areas" sets both the width for area 5 and height for areas 4 and 6.

Three tendencies [increment (+), decrement (−), and equal (=)] are possible for the y-coordinate of a point after the rotation depending on the area the point belongs to (Fig. 6).

The "voting" step ends when each pair of corresponding points vote. There are five voting actions considered, as Table 1 resumes. Voting actions are selected taking into account the sign of the vertical parallax between the pair of corresponding points, the area each corresponding point belongs to, and the axis being considered.

Given a pair of corresponding points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, where the y vertical parallax is positive, $(y_2 - y_1) > 0$. If P_1 belongs to the area 1 (left upper quadrant, Fig. 5), and the point is asked about a positive rotation around X-axis, the answer will be "no" [Fig. 6(a)]. But if the point is asked about a negative rotation around the same X-axis, the answer will be "yes." This point would give the same vote for a positive rotation around Y-axis and the opposite answer if it was asked about a positive rotation around Z-axis [Figs. 6(b) and 6(c), respectively]. These voting actions, "yes" and "no," are named vote I and vote II, respectively (Fig. 7 and Table 1).

If the vertical parallax between $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is lower than the "MaxYError" threshold, the votes of P_1

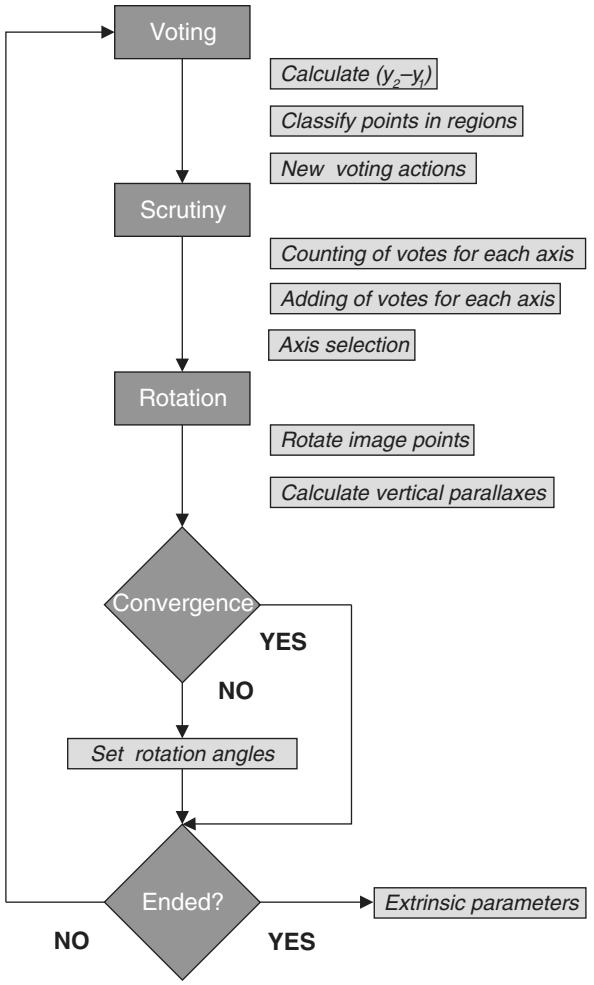

Fig. 4 Conceptual algorithm description.

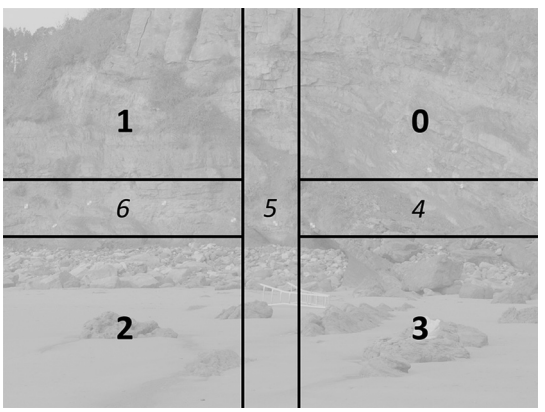

Fig. 5 Main (bold) and minor (italics) areas considered in each image.

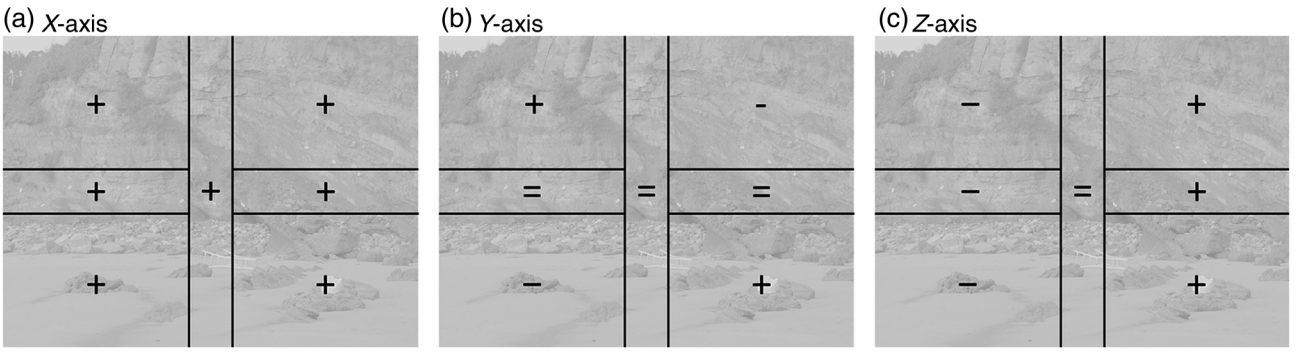

Fig. 6 Modification of *y*-coordinate after a positive rotation around axes (a) *X*, (b) *Y*, and (c) *Z* regarding the area the point belongs to.

Table 1 Voting actions considered.

Vote	Meaning
I	A positive rotation around this axis is good for this point
II	A negative rotation around this axis is good for this point
III	Any rotation around this axis has to be avoided, the vertical parallax is low enough (this is a "stop point"), and this rotation would increase/decrease it
IV	This rotation does not affect the vertical parallax, but the vertical parallax of this point is fortunately low (this is a "stop point")
V	This rotation does not affect the vertical parallax, but the vertical parallax is regrettably high for this point

when asked about a rotation around any axis would be "no." This voting action is named vote III (Table 1).

If P_1, a point belonging to area 6 (Fig. 5), is asked about a rotation around the *Y*-axis, the answer will be "I do not care," since any rotation around *Y*-axis does not affect the corresponding point's vertical parallax [Fig. 3(b)]. Depending on the actual vertical parallax, the vote may be IV (when the value is below the "MaxYError") or V (when it is too high that the rotation will not solve the problem) (Table 1).

Once each pair of corresponding points has emitted its vote, the "scrutiny" starts (Fig. 4). The five types of voting actions (votes) emitted are grouped into two categories for each axis and photograph, i.e., positive or negative rotation around the axis. The formulas used in the current implementation of the algorithm are

$$\text{positive rot}_{\text{axis}}^{\text{photo}} = \sum_{i=0}^{2n} [(\text{vote I} + \text{vote IV}) - (\text{vote II} + \text{vote III} + \text{vote V})]_i, \tag{6}$$

$$\text{negative rot}_{\text{axis}}^{\text{photo}} = \sum_{i=0}^{2n} [(\text{vote II} + \text{vote IV}) - (\text{vote I} + \text{vote III} + \text{vote V})]_i, \tag{7}$$

where "n" denotes the number of corresponding pairs. Note that vote III and vote V have the same effect in the final result. Nevertheless, the distinction between both voting

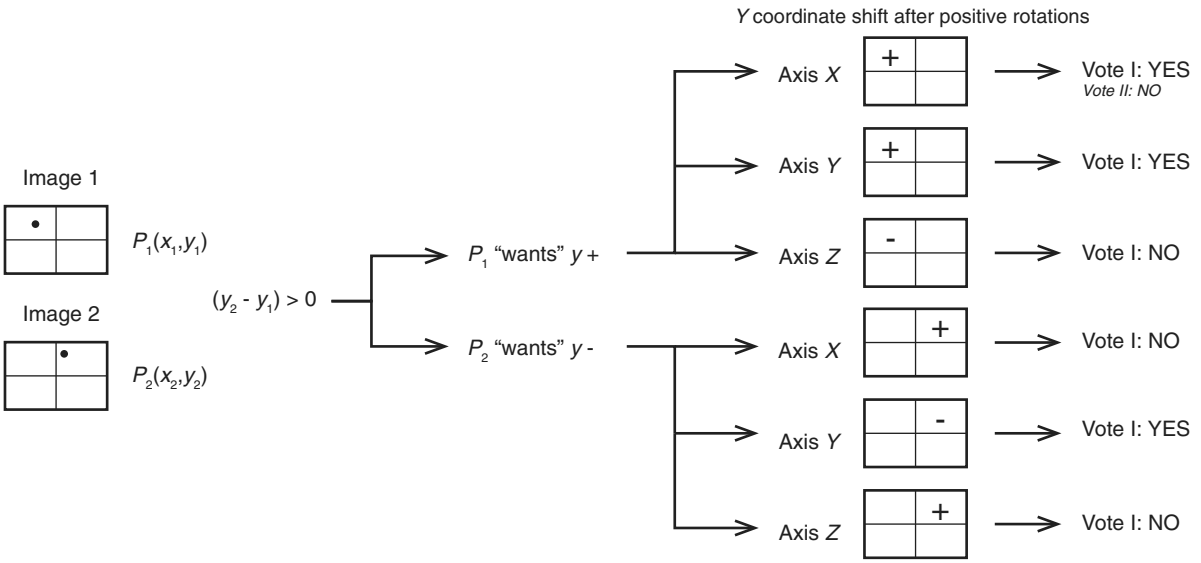

Fig. 7 Diagram illustrating the voting process.

actions is maintained: it opens the possibility of modifying the algorithm in the future, e.g., setting different weights for each vote. At this point, there are three axes, two photos, and two opposite directions, i.e., twelve possible solutions. Each alternative ("rotation") has a value associated, thanks to the scrutiny [Eqs. (6) and (7)]. That value can even be negative, as there are negative votes in both Eqs. (6) and (7). The choice can be deterministic, i.e., just the rotation with more votes (although draws are considered); or probabilistic, using a formula to select the final choice with a randomized variable. The current implementation of the algorithm uses a probabilistic approach: with a probability of 95%, a rotation with positive votes will be selected (parameter "vote + weight"). In that case, the probability of selecting one rotation is proportional to the relative number of votes counted for that choice.

Next, the rotation selected is applied ("rotation" in Fig. 4). The corresponding points are calculated. As a rotation angle differently affects the y vertical parallax depending on the rotation axis, a base angle is defined for each axis (parameters "baseX," "baseY," and "baseZ"). Once the rotation is applied, the y vertical parallax is recalculated. Then, the sum of vertical parallaxes is calculated. Furthermore, the number of corresponding points with vertical parallax lower than a threshold (known as "stop points") is calculated.

The last step is called "evaluation" (Fig. 4). Whether the sum of vertical parallaxes is lower after the rotation or the number of "stop points" is increased, the algorithm will converge. On the contrary, if the algorithm is not converging, the rotation values will be discarded.

The current implementation of the algorithm uses two different strategies to avoid convergence problems. First, thanks to the probabilistic approach used in the "scrutiny," a different angle can be selected in the next iteration to fulfill convergence. Second, if one axis fails to produce a convergence result up to "t" trials, the base rotation angle for that axis is reduced in a percentage. This reduction is always needed during the last iterations to increase the accuracy of the method.

Three different scenarios are considered to stop the iteration process: first, all the correspondence points are classified as "stop points;" second, all rotation axes have received negative votes during the last "n" iterations; and third, the iteration counter reaches the maximum number of iterations.

3 Results

The heuristic method based on voting was implemented in C# using Microsoft Visual Studio 2010 to test its performance and accuracy. Two case studies are presented: the first deals with a simulated (synthetic) stereopair and the second with a true stereopair taken with a calibrated camera. In order to compare the results with a different and well-known rectification algorithm, the relative orientation results using the nonlinear least-squares adjustment based on the coplanarity condition are provided, too.

3.1 *Simulated Case Study*

An ideal simulated scene with 121 corresponding points is mathematically recreated (Fig. 8). Both central areas of the images (which comprise the 50% in width and height of the surface), before and after the rectification, are covered by corresponding points.

The virtual camera has no distortion and its intrinsic matrix M is defined as

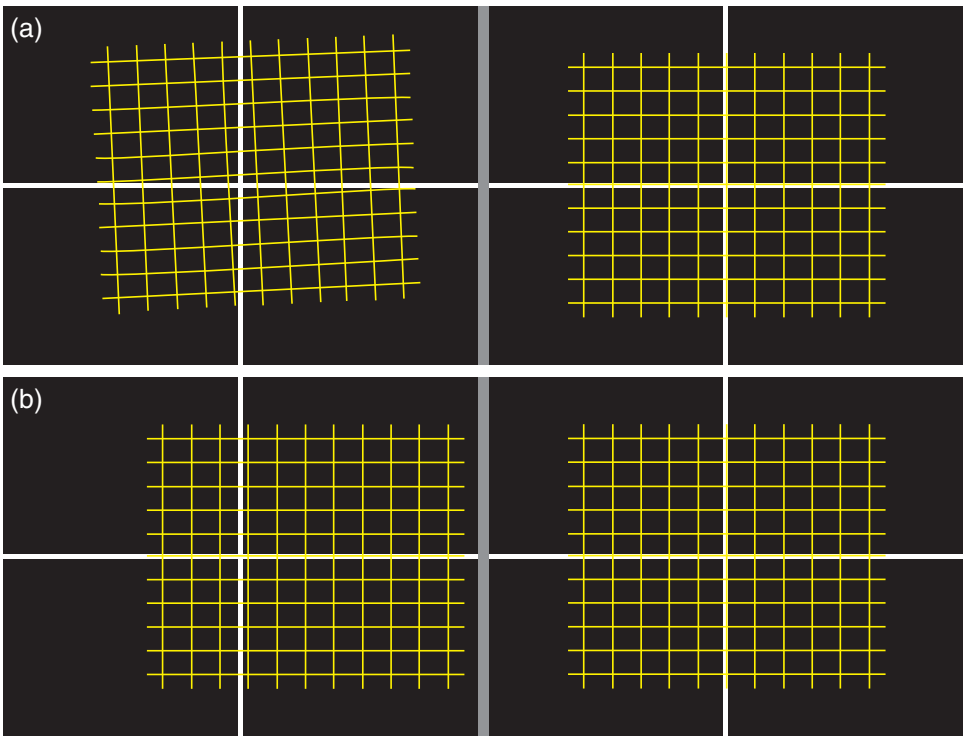

Fig. 8 Corresponding points on the stereopair used for the simulated case study: (a) input dataset, i.e., initial stereopair without epipolarization and (b) epipolarized stereopair.

Table 2 Algorithm configuration parameters.

MaxYError (pixels)	1
Axial areas (pixels)	36
BaseX (deg)	0.03
BaseY (deg)	0.70
BaseZ (deg)	0.13
Vote + weight (%)	95

$$M = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 7500 & 0 & 1824 \\ 0 & 7500 & 1368 \\ 0 & 0 & 1 \end{pmatrix}.$$

The object points cover a rectangle grid of 2200×1800 units and are 7500 units away from the cameras. The distance between both projection centers is 500 units in X-axis direction. The rotation between both cameras, expressed using Rodrigues' rotation vector is

$$r = (\,0.010 \quad -0.050 \quad -0.040\,).$$

This rotation affects only the left camera in this simulated case study. In this notation, the module of r defines the

Table 3 Simulation results achieved using the nonlinear least-squares relative orientation based on the coplanarity condition and the heuristic method (15 trials).

		Mean error		Points error	Rodrigues' rotation result			Translation result		
Trial	Iter.	$\lvert y_2' - y_1' \rvert$	SD	<1 pixels	X	Y	Z	X	Y	Z
				Simulation values						
					0.010	−0.050	−0.040	−1.000	0.000	0.000
				Nonlinear least-squares based on the coplanarity condition						
		0.000	0.000	121	0.011	−0.056	−0.044	−1.000	−0.084	0.007
				Heuristic method						
1	73	0.308	0.336	121	0.010	−0.055	−0.041	−1.000	−0.016	0.001
2	66	0.335	0.231	121	0.010	−0.049	−0.041	−1.000	−0.023	0.000
3	70	0.503	0.351	121	0.011	−0.049	−0.041	−1.000	−0.013	0.012
4	72	0.535	0.423	121	0.011	−0.049	−0.041	−1.000	−0.022	0.012
5	83	0.589	0.395	104	0.011	−0.048	−0.040	−1.000	−0.016	0.011
6	62	0.696	0.493	104	0.011	−0.049	−0.040	−1.000	−0.012	0.012
7	68	0.803	0.815	99	0.010	−0.037	−0.039	−1.000	−0.010	0.000
8	77	0.815	0.754	82	0.011	−0.037	−0.040	−1.000	−0.017	0.012
9	73	0.825	0.568	80	0.011	−0.049	−0.039	−1.000	−0.022	0.012
10	65	0.833	0.485	55	0.011	−0.049	−0.040	−1.000	−0.017	0.025
11	67	0.837	0.485	64	0.011	−0.049	−0.040	−1.000	−0.011	0.024
12	79	0.870	0.955	73	0.011	−0.037	−0.041	−1.000	−0.022	0.012
13	80	0.929	0.627	71	0.011	−0.048	−0.039	−1.000	−0.015	0.024
14	68	1.047	0.665	65	0.011	−0.045	−0.039	−1.000	−0.019	0.020
15	73	1.365	1.106	56	0.012	−0.035	−0.040	−0.999	−0.014	0.035

Number of corresponding points, 121; initial mean error, 87.731 pixels; SD, standard deviation; and Iter., number of iterations for each run.

magnitude of the rotation angle (in radians), and its direction defines the axis of rotation. For this example, the rotation angle is equal to 3.713 deg.

As the solution of the problem ignores the scale of the scene, the normalized expected results will be a translation vector $t = (-1,0,0)$ and a rotation vector $r = (0.010, -0.050, -0.040)$ using right camera as reference.

In order to compare the results, the problem is solved using both the heuristic method and relative orientation based on the coplanarity condition. Table 2 presents the *a priori* configuration parameters and thresholds set for the heuristic method. The relative orientation based on the coplanarity condition works with no rotation on the left camera and a fixed X value for the translation vector (when referred to the left camera).

Table 3 compares the expected results with those achieved with the nonlinear least-squares relative orientation, as well as with the heuristic method (in this case after running 15 trials). The total number of iterations, mean error, standard deviation, and results obtained are registered. The total number of correspondence points with a final vertical parallax error lower than 1 pixel are also presented for each trial. Trials are sorted to show best results on top. All but 1 out of the 15 trials ended because no axis received positive votes during the last iterations, trial 2 stopped because the absolute vertical parallax of all corresponding points was <1 pixel. The final mean error is <1 pixel in 13 out of 15 trials. The nonlinear least-squares relative orientation with rotations based on the coplanarity condition provides almost null mean error. As stated before, the nonlinear least-squares adjustment is robust and accurate when there are no local (false) minima, as expected in a simulated case.

Figure 9 shows the tendency of one trial after 64 iterations (this trial is not included in Table 3). The error, measured as the mean absolute difference $|y_2 - y_1|$ between the 121 corresponding points, decreases from 87.731 to 0.204 pixel in this trial. The number of points within a difference $|y_2 - y_1| < 1$ pixel ("stop points") reaches 121, i.e., the total number of points. Additionally, the total number of votes obtained for the best axis is also represented: no axis received positive votes after 62 iterations.

In this simulation, a Dell Precision T5610 workstation was used. A typical trial, 70 iterations and 121 corresponding points, is solved in about 14 ms.

3.2 *Real Case Study*

A stereopair of digital photographs was used for this case study. Photographs were taken with a digital single-lens reflex camera Olympus E-410 (3648×3726 pixels), using a ZUIKO DIGITAL 35-mm F3.5 macrolens (focal length equivalent to 70 mm on a 35-mm camera). The camera was calibrated using a self-developed software, which uses the Open Source Computer Vision (OpenCV) library.[3] Only two radial distortion terms were considered, k_1 and k_2, and both tangential distortion coefficients p_1 and p_2 were set to zero. The intrinsic camera matrix and distortion coefficients obtained were

$$M = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 7509.285 & 0 & 1832.675 \\ 0 & 7508.130 & 1451.393 \\ 0 & 0 & 1 \end{pmatrix},$$

$$C = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} -0.197365 \\ 0.900129 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The pair of photographs was used in a previous geological research. The speeded up robust features (SURF) algorithm[23] was used to obtain the corresponding features. The Hessian threshold was set to 600, whereas the match distance ratio

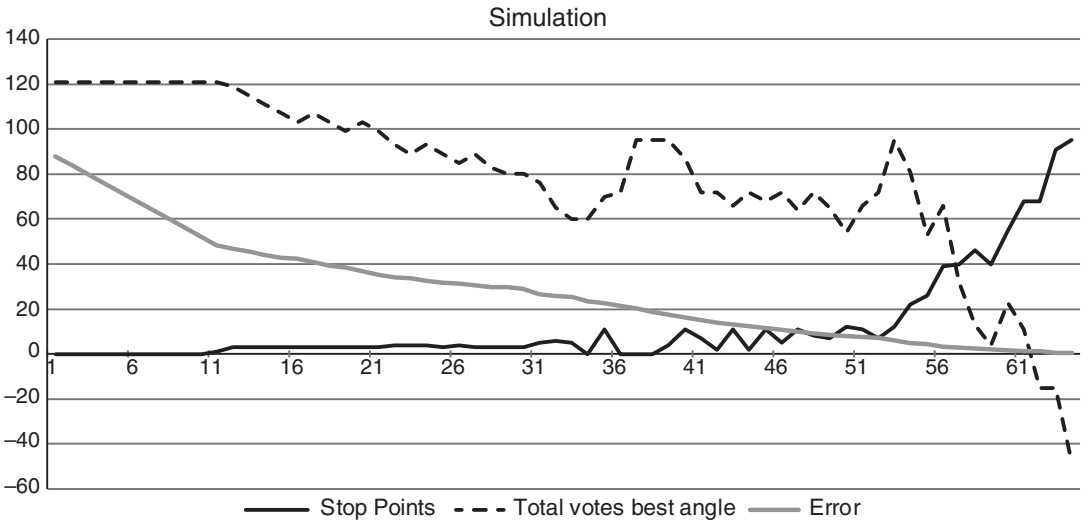

Fig. 9 Number of stop points; total number of votes received by best axis; and mean absolute parallax (error) per iteration for a trial of the simulated stereopair presented in Fig. 8.

Fig. 10 Corresponding points obtained after SURF on the geological stereo-pair case study.

was set to 0.3. As a result, 298 corresponding points were detected (Fig. 10). The distribution of the points was accepted initially, although the lower half of the image had a small number of points.

As before, the solution obtained with the heuristic method is compared to the one offered by the nonlinear least-squares relative orientation with rotations based on the coplanarity condition. Table 4 compares the results achieved with

Table 4 Results achieved with a true stereopair.

Trial	Iter.	Mean error $\lvert y'_2 - y'_1 \rvert$	SD	Points error <1 pixels	Rodrigues' rotation result X	Y	Z	Translation result X	Y	Z
Nonlinear least-squares based on the coplanarity condition										
		0.304	0.467	283	0.007	−0.102	−0.021	−1.000	−0.020	0.087
Heuristic method										
1	63	0.448	0.501	271	0.006	−0.084	−0.019	−0.997	−0.012	0.072
2	67	0.613	0.546	234	0.006	−0.086	−0.020	−0.998	−0.017	0.063
3	77	0.779	0.928	219	0.006	−0.077	−0.018	−0.998	−0.010	0.061
4	57	0.816	0.640	214	0.006	−0.082	−0.019	−0.998	−0.011	0.060
5	64	0.854	0.913	196	0.006	−0.073	−0.018	−0.998	−0.007	0.062
6	58	0.902	1.033	208	0.006	−0.084	−0.020	−0.998	−0.055	0.060
7	147	0.932	0.828	201	0.006	−0.078	−0.019	−0.999	−0.012	0.053
8	63	0.988	1.234	214	0.006	−0.075	−0.018	−0.998	−0.007	0.056
9	56	1.092	0.932	182	0.006	−0.069	−0.018	−0.998	−0.008	0.057
10	57	1.098	0.991	173	0.005	−0.075	−0.019	−0.999	−0.011	0.044
11	64	1.145	1.162	176	0.006	−0.069	−0.018	−0.999	−0.007	0.046
12	115	1.294	1.061	141	0.005	−0.080	−0.018	−0.999	−0.011	0.043
13	60	1.407	1.334	152	0.005	−0.065	−0.019	−0.999	−0.014	0.031
14	71	1.563	1.272	116	0.005	−0.068	−0.019	−0.999	−0.013	0.031
15	150	1.997	1.879	117	0.005	−0.067	−0.019	−1.000	−0.012	0.013

Number of corresponding points, 298; initial mean error, 43.960 pixels; SD, standard deviation; and Iter., number of total iterations for each run.

each method (after running 15 trials in the case of the heuristic one). A maximum of 150 iterations was set for the heuristic technique. The total number of iterations per trial, error and standard deviation, number of stop points, and results obtained are presented in Table 4. 14 out of 15 trials ended because no axis received positive votes during the last iterations; only 1 finished after 150 iterations. The final mean error is <1 pixel in 8 out of 15 trials. The best trial, number 1, stopped after 63 iterations with a mean error of 0.448 pixel; 271 out of 298 corresponding points had a vertical final parallax lower than 1 pixel. The nonlinear least-squares relative orientation with rotations based on the coplanarity condition has provided a slightly lower mean error of 0.304 pixel, with 283 out of 298 corresponding points with parallax lower than 1 pixel.

4 Discussion

The results obtained in both case studies confirm that it is possible to obtain a final mean error below 1 pixel. This result was achieved in 13 out of 15 trials in the simulated case and in 8 out of 15 trials in the real one; the rest were also close to 1. In the best simulated case, all 121 corresponding points had a final error <1 pixel; whereas in the real case it was achieved for 271 of 288 points. Ideally, as there should not be any vertical parallax between corresponding points in an epipolarized stereopair, this is the evidence of success of the presented heuristic method based on voting.

The performance (repeatability) of the method can be measured using the standard deviation of the mean error obtained on those 15 trials. The simulated case study shows a mean value of the mean error $|y'_2 - y'_1|$ equal to 0.753 pixel, with a standard deviation of 0.579 pixel; whereas the real case study has a mean value of 1.062 pixel with a standard deviation of 1.017 pixel. Those values are lower enough to guarantee the consistency of the algorithm.

The simulation example gives us another measurement of the performance of the method as far as the accuracy is concerned, as the rotation is known *a priori*. The three components of Rodrigues' rotation vector fit the expected value up to the second decimal in 10 out of 15 trials.

The nonlinear least-squares relative orientation method with rotations based on the coplanarity condition provides almost null mean error in the simulated case and a slightly lower mean error in the real one. This method leads to satisfactory convergence when started from an approximate estimate of camera poses (true in both the real and simulated examples used here) and yields precise results when there are no false minima (especially true in the simulated case).

It is worth noticing that the nonlinear least-squares relative orientation method based on the collinearity condition did not work out the determination of the extrinsic orientation parameters. Because of this, the results of this alternative solution of the camera pose were not presented to check the performance of the heuristic method proposed in this paper.

The results of the heuristic method are good enough to consider this technique as an alternative or complement to the classical approach relative orientation, due to its robustness to handle outliers. In future works, the strengths and weaknesses of both methods should be tested in depth in different scenarios.

The algorithm configuration parameters used in both examples are summarized in Table 2. The "MaxYError" threshold is set to 1 pixel. This value affects not only the criteria to label a pair of corresponding points as "StopPoint" but also the meaning of the votes emitted by this pair, as explained in Table 1. When voting a rotation axis that does not affect the vertical parallax, the vote of a pair of points with error above the "MaxYError" will be vote IV. But if "MaxYError" is increased and the corresponding points are labeled as "StopPoint," the vote meaning will be vote V. Both votes have exactly the opposite effect. The "StopPoint" tendency is to facilitate rotations in axes that do not affect vertical parallax, whereas non-"StopPoint" tendency is to block those rotations.

"Central areas" threshold is indirectly correlated with "MaxYError," because both have direct influence in votes IV and V. As "central areas" threshold increases, the number of points lying in areas 4, 5, and 6 will increase (Fig. 5). Those areas, next to both horizontal and vertical axes, are likely to produce less effect in parallax when rotating around *Y*- or *Z*-axes (Fig. 6). The lower the "central areas" value, the less votes IV and V. The value adopted in both examples is 36 pixels (Table 2).

The parameters "baseX," "baseY," and "baseZ" are calculated taking into account Fig. 3. Those angles produce a vertical parallax variation lower than 4 pixels. The higher those values are, the faster the algorithm evolves during the first iterations. But as the algorithm converges to the solution, the angles will be reduced in order to achieve a lower error.

"Vote + weight" represents the probability of an axis with positive votes to be selected. If it was increased to 100%, there could be convergence problems. This random effect during the "scrutiny" step is desirable, i.e., no axis receives positive votes after 62 iterations in a trial of the simulated problem (Fig. 9), but the error can still be reduced significantly during the last three iterations, from 0.657 to 0.204 pixel. The value adopted in both examples is 95% (Table 2).

Concerning the speed of the method proposed, in both examples the mean number of iterations is lower than 100. As said previously, the number of iterations can be reduced if the base angles values are increased. But, on the other hand, it can produce convergence issues. The "voting" step, according to Fig. 4, is parallelizable, taking advantage of the computer architecture.

The code of the algorithm is written in C#, and it is not optimized. However, a typical trial of 70 iterations with 121 corresponding points is solved in just about 14 ms using a Dell Precision T5610 workstation. In its present form, the algorithm is very robust due to the voting strategy. Outliers can be easily detected and removed from the final solution due to the comparatively high vertical parallax error achieved after several iterations.

5 Conclusions

The heuristic method based on voting proposed herein solves both the extrinsic orientation problem and the image rectification in just one single step. A set of corresponding points, plus the intrinsic (interior) orientation parameters, including the lens distortion parameters, are given to the algorithm. The result will be two rotation matrices, one for the left image and another for the right, which minimize the vertical parallaxes

between corresponding points. The rotation matrices can be combined into a single rotation matrix between the two cameras.

The algorithm performance and accuracy are checked using both simulated and real case studies and compared with the traditional nonlinear least-squares relative orientation with rotations based on the coplanarity condition; the relative orientation based on the collinearity condition did not work out. No *a priori* knowledge of the extrinsic parameters is required. The heuristic method based on voting is robust, fast, and yields optimal results as far as there are common points in the main areas of the images. Further research will be devoted to determine estimates of the method under unconventional stereo setups.

Acknowledgments

The authors gratefully acknowledge the support from the Spanish Ministerio de Economía y Competitividad to the Project No. HAR2014-59873-R.

References

1. H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from projections," *Nature* **293**, 133–135 (1981).
2. Y. Ma et al., *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer Verlag, New York (2003).
3. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Press, Cambridge (2008).
4. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge (2004).
5. T. Luhmann et al., *Close-Range Photogrammetry and 3D Imaging*, De Gruyter, Berlin (2013).
6. K. Kraus, *Photogrammetry: Geometry from Images and Laser Scans*, De Gruyter, Berlin (2007).
7. J. L. L. García, *Fotogrametría Moderna: Analítica y Digital*, Universidad Politécnica de Valencia, Valencia (2002).
8. B. Triggs et al., "Bundle adjustment—a modern synthesis," *Lect. Notes Comput. Sci.* **1883**, 298–372 (2000).
9. J. C. McGlone et al., *Manual of Photogrammetry*, 5th ed., American Society of Photogrammetry and Remote Sensing, Virginia (2004).
10. H. Stewénius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS J. Photogramm. Remote Sens.* **60**, 284–294 (2006).
11. E. Kruppa, "Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung," *Sitz.-Ber. Akad. Wiss. Wien. Math. Naturwiss. Kl. Abt. IIa.* **122**, 1939–1948 (1913).
12. O. Pizarro, R. Eustice, and H. Singh, "Relative pose estimation for instrumented, calibrated imaging platforms," in *Proc. of VII Digital Image Computing Techniques and Applications*, pp. 601–612 (2003).
13. A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo-pairs," *Mach. Vision Appl.* **12**, 16–22 (2000).
14. X. Song, L. D. Seneviratne, and K. Althoefer, "A Kalman filter-integrated optical flow method for velocity sensing of mobile robots," *IEEE/ASME Trans. Mechatron.* **16**, 551–563 (2011).
15. G. Panahandeh and M. Jansson, "Vision-aided inertial navigation based on ground plane feature detection," *IEEE/ASME Trans. Mechatron.* **19**, 1206–1215 (2014).
16. M. Caccia, "Vision-based ROV horizontal motion control: near-seafloor experimental results," *Control Eng. Pract.* **15**, 703–714 (2007).
17. M. R. Halfawy and J. Hengmeechai, "Optical flow techniques for estimation of camera motion parameters in sewer closed circuit television inspection videos," *Autom. Constr.* **38**, 39–45 (2014).
18. C.-K. Tang, G. Medioni, and M.-S. Lee, "N-dimensional tensor voting and application to epipolar geometry estimation," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 829–844 (2001).
19. W.-S. Tong, C.-K. Tang, and G. Medioni, "Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1167–1184 (2004).
20. S. Obdržálek and J. Matas, "A voting strategy for visual ego-motion from stereo," in *Proc. of IEEE Intelligent Vehicles Symp.*, pp. 382–387 (2010).
21. Y. Yuan et al., "Efficient image matching using weighted voting," *Pattern Recognit. Lett.* **33**, 471–475 (2012).
22. T. Wu et al., "Automatic cloud detection for high resolution satellite stereo images and its application in terrain extraction," *ISPRS J. Photogramm. Remote Sens.* **121**, 143–156 (2016).
23. H. Bay et al., "SURF: speeded up robust features," *Comput. Vision Image Understanding* **110**, 346–359 (2008).

Santiago Martín received his PhD in industrial engineering from the University of Oviedo in 1997. He worked as an environmental specialist during seven years in private companies and the Asturias regional government. Since 2003 he has been a professor of the University of Oviedo, teaching several subjects related to engineering graphics. He works on virtual reality, computer graphics, and computer vision. He is interested in the application of them to fields such as geology.

José Luis Lerma has been teaching photogrammetry related topics in the Universitat Politècnica de València, where he obtained his PhD in geodesy and cartography in 1999, and other institutions since 1996. In addition to this, he has been a consultant on different projects focused mainly on the use of photogrammetry for the documentation and preservation of cultural heritage.

Hodei Uzkeda received his PhD in geology from the University of Oviedo in 2013, where he is currently working as a postdoctoral researcher. Since his PhD studies he has been interested on the employment of photogrammetry to the creation of virtual outcrop models as an essential tool to obtain accurate and reliable geological information.