(will be inserted by hand later)

A Compact Algorithm for Rectification of Stereo Pairs

Andrea Fusiello[1], **Emanuele Trucco**[2], **and Alessandro Verri**[3]

[1] Dipartimento di Matematica e Informatica, Università di Udine, Via delle Scienze, 206, I-33100 Udine,IT
Phone: +39 0432 558455, Fax: +39 0432 558499, e-mail fusiello@dimi.uniud.it
[2] Heriot-Watt University, Department of Computing and Electrical Engineering, Edinburgh, UK
[3] Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Genova, IT

January 16, 1999

Abstract. We present a linear rectification algorithm for general, unconstrained stereo rigs. The algorithm takes the two perspective projection matrices of the original cameras, and computes a pair of rectifying projection matrices. It is compact (22-line MATLAB code) and easily reproducible. To maximize usefulness, we have made its implementation available on the WWW. We report tests proving the correct behavior of our method, as well as the negligible decrease of the accuracy of 3-D reconstruction performed from the rectified images directly.

Key words: stereo, epipolar geometry, rectification

1 Introduction and motivations

Given a pair of stereo images, *rectification* determines a transformation of each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes (usually the horizontal one). The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras. The important advantage of rectification is that computing stereo correspondences (Dhond and Aggarwal, 1989; Fusiello et al., 1997) is made simpler, because search is done along the horizontal raster lines of the rectified images.

We assume that the stereo rig is *calibrated*, i.e., cameras' internal parameters and their mutual position and orientation are known. This assumption is not strictly necessary, but leads to a simpler technique. On the other hand, when reconstructing 3-D shape of objects from dense stereo, calibration is mandatory in practice, and can be achieved in many situations and by several algorithms (Caprile and Torre, 1990; Robert, 1996).

Rectification is a classical problem of stereo vision; however, few methods are available in the Computer Vision literature, to our knowledge. Ayache (Ayache and Lustman, 1991) introduced a rectification algorithm, in which a matrix satisfying a number of constraints is hand-crafted. The distinction between necessary and arbitrary constraints is unclear. Some authors report rectification under restrictive assumptions; for instance, (Papadimitriou and Dennis, 1996) assumes a very restrictive geometry (parallel vertical axes of the camera reference frames). Recently, (Hartley and Gupta, 1993; Robert et al., 1997; Hartley, 1998) have introduced an algorithm which performs rectification given a *weakly calibrated* stereo rig (i.e., a rig for which only points correspondences between images are given).

This paper presents a novel algorithm rectifying a *calibrated* stereo rig of *unconstrained geometry* and mounting general cameras. Our work improves and extends (Ayache and Lustman, 1991). We obtain basically the same results, but in a more compact and clear way. The algorithm is simple and detailed. It is first derived following a simple geometrical argument, and it is then proven correct analytically. Moreover, given the shortage of easily reproducible, easily accessible and clearly stated algorithms we have made a "rectification kit" (code, examples and instructions) available on the WWW.

This paper is organized as follows. Section 2 introduces our notations and summarizes some necessary mathematics of perspective projections. Section 3 derives the algorithm for computing the rectifying PPMs and Section 4 expresses the rectifying image transformation in terms of PPMs. Section 5 gives the compact (22 lines), working MATLAB code for our algorithm. Section 6 reports tests on synthetic and real data. Section 7 is a brief discussion of our work. Appendix A gives a formal proof of the correctness of the algorithm.

2 Camera model and epipolar geometry

This section briefly recalls the mathematical background on perspective projections necessary for our purposes. For more details see (Faugeras, 1993).

Correspondence to: Andrea Fusiello

2.1 Camera model

A pinhole camera is modeled by its *optical center* C and its *retinal plane* (or *image plane*) $\mathcal{R}$. A 3-D point W is projected into an image point M given by the intersection of $\mathcal{R}$ with the line containing C and W (Figure 1). The line containing C and orthogonal to $\mathcal{R}$ is called the *optical axis* (the Z axis in Figure 1) and its intersection with $\mathcal{R}$ is the *principal point*. The distance between C and $\mathcal{R}$ is the *focal distance* (note that, since in this model C is behind $\mathcal{R}$, real cameras will have negative focal distance).

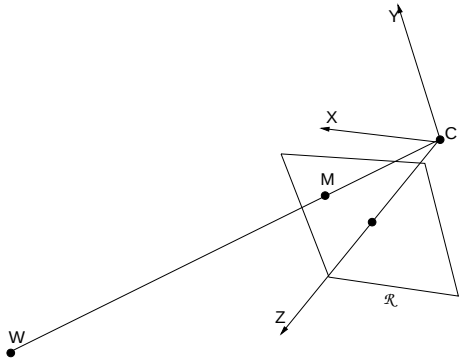


Fig. 1. The pinhole camera model, with the *camera reference frame* (X,Y,Z) depicted.

Let $\mathbf{w} = [x \;\; y \;\; z]^\top$ the coordinates of W in the world reference frame (fixed arbitrarily) and $\mathbf{m} = [u \; v]^\top$ the coordinates of M in the image plane (pixels). The mapping from 3-D coordinates to 2-D coordinates is the *perspective projection*, which is represented by a linear transformation in *homogeneous coordinates*. Let $\tilde{\mathbf{m}} = [u \; v \; 1]^\top$ and $\tilde{\mathbf{w}} = [x \; y \; z \; 1]^\top$ be the homogeneous coordinates of M and W respectively; then the perspective transformation is given by the matrix $\tilde{\mathbf{P}}$:

$$\lambda \tilde{\mathbf{m}} = \tilde{\mathbf{P}} \tilde{\mathbf{w}}, \tag{1}$$

where λ is an arbitrary scale factor. The camera is therefore modeled by its *perspective projection matrix* (henceforth PPM) $\tilde{\mathbf{P}}$, which can be decomposed, using the QR factorization, into the product

$$\tilde{\mathbf{P}} = \mathbf{A}[\mathbf{R} \mid \mathbf{t}]. \tag{2}$$

The matrix $\mathbf{A}$ depends on the *intrinsic parameters* only, and has the following form:

$$\mathbf{A} = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\alpha_u = -fk_u$, $\alpha_v = -fk_v$ are the focal lengths in horizontal and vertical pixels, respectively (f is the focal length in millimeters, k_u and k_v are the effective number of pixels per millimeter along the u and v axes), (u_0, v_0) are the coordinates of the *principal point*, given by the intersection of the optical axis with the retinal plane, and γ is the *skew* factor that models non-rectangular CCD elements.

The camera position and orientation (*extrinsic parameters*), are encoded by the 3×3 rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$, representing the rigid transformation that brings the camera reference frame onto the world reference frame.

Let us write the PPM as

$$\tilde{\mathbf{P}} = \left[\begin{array}{c|c} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{array}\right] = [\mathbf{Q}|\tilde{\mathbf{q}}]. \tag{4}$$

In Cartesian coordinates, the projection (1) writes

$$\begin{cases} u = \dfrac{\mathbf{q}_1^\top \mathbf{w} + q_{14}}{\mathbf{q}_3^\top \mathbf{w} + q_{34}} \\ v = \dfrac{\mathbf{q}_2^\top \mathbf{w} + q_{24}}{\mathbf{q}_3^\top \mathbf{w} + q_{34}}. \end{cases} \tag{5}$$

The *focal plane* is the plane parallel to the retinal plane that contains the optical center C (the plane XY in Fig. 1). The coordinates $\mathbf{c}$ of C are given by

$$\mathbf{c} = -\mathbf{Q}^{-1}\tilde{\mathbf{q}}. \tag{6}$$

Therefore $\tilde{\mathbf{P}}$ can be written:

$$\tilde{\mathbf{P}} = [\mathbf{Q}| - \mathbf{Q}\mathbf{c}]. \tag{7}$$

The *optical ray* associated to an image point M is the line M C, i.e. the set of 3-D points $\{\mathbf{w} : \tilde{\mathbf{m}} = \tilde{\mathbf{P}}\tilde{\mathbf{w}}\}$. The equation of this ray can be written in parametric form as

$$\mathbf{w} = \mathbf{c} + \lambda \mathbf{Q}^{-1}\tilde{\mathbf{m}} \quad \lambda \in R. \tag{8}$$

2.2 Epipolar geometry

Let us consider a stereo rig composed by two pinhole cameras (Fig. 2). Let C_1 and C_2 be the optical centers of the left and right cameras respectively. A 3-D point W is projected onto both image planes, to points M_1 and M_2, which constitute a conjugate pair. Given a point M_1 in the left image plane, its conjugate point in the right image is constrained to lie on a line called the *epipolar line* (of M_1). Since M_1 may be the projection of an arbitrary point on its optical ray, the epipolar line is the projection through C_2 of the optical ray of M_1. All the epipolar lines in one image plane pass through a common point (E_1 and E_2 respectively) called the *epipole*, which is the projection of the conjugate optical center.

When C_1 is in the focal plane of the right camera, the right epipole is at infinity, and the epipolar lines form a bundle of parallel lines in the right image. A very special case is when both epipoles are at infinity, that happens when the line $\mathsf{C}_1\mathsf{C}_2$ (the *baseline*) is contained in both focal planes, i.e., the retinal planes are parallel to the baseline. Epipolar lines, then, form a bundle of parallel lines in both images. Any pair of images can be transformed so that epipolar lines are parallel and horizontal in each image. This procedure is called *rectification*.

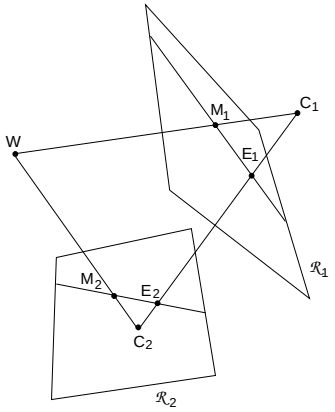


Fig. 2. Epipolar geometry. The epipole of the first camera E is the projection of the optical center C_2 of the second camera (and vice versa).

3 Rectification of camera matrices

We assume that the stereo rig is *calibrated*, i.e., the old PPMs $\tilde{\mathbf{P}}_{o1}$ and $\tilde{\mathbf{P}}_{o2}$ are known. The idea behind rectification is to define two new PPMs $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ obtained by rotating the old ones around their optical centers until focal planes becomes coplanar, thereby containing the baseline. This ensures that epipoles are at infinity, hence epipolar lines are *parallel*. To have *horizontal* epipolar lines, the baseline must be parallel to the new X axis of both cameras. In addition, to have a proper rectification, conjugate points must have the *same vertical coordinate*. This is obtained by requiring that the new cameras have the same intrinsic parameters. Note that, being the focal distance the same, retinal planes are coplanar too, as in Figure 3.

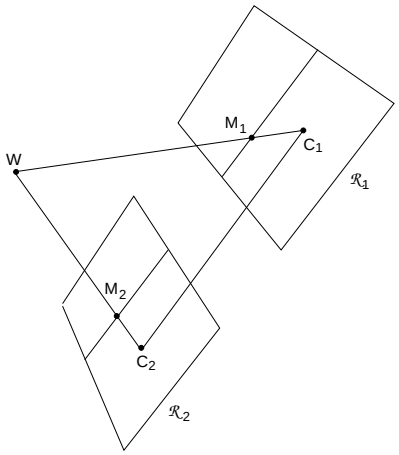


Fig. 3. Rectified cameras. Retinal planes are coplanar and parallel to the baseline.

In summary: positions (i.e, optical centers) of the new PPMs are the same as the old cameras, whereas the new orientation (the same for both cameras) differs from the old ones by suitable rotations; intrinsic parameters are the same for both camera. Therefore, the two resulting PPMs will differ only in their optical centers, and they can be thought as a single camera translated along the X axis of its reference system.

Let us write the new PPMs in terms of their factorization. From (2) and (7):

$$\tilde{\mathbf{P}}_{n1} = \mathbf{A}[\mathbf{R} \mid -\mathbf{R}\,\mathbf{c}_1], \quad \tilde{\mathbf{P}}_{n2} = \mathbf{A}[\mathbf{R} \mid -\mathbf{R}\,\mathbf{c}_2]. \quad (9)$$

The intrinsic parameters matrix $\mathbf{A}$ is the same for both PPMs, and can be chosen arbitrarily (see MATLAB code). The optical centers $\mathbf{c}_1$ and $\mathbf{c}_2$ are given by the old optical centers, computed with (6). The matrix $\mathbf{R}$, which gives the camera's pose, is the same for both PPMs. It will be specified by means of its row vectors

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} \quad (10)$$

that are the X, Y, and Z axes, respectively, of the camera reference frame, expressed in world coordinates.

According to the previous comments, we take:

1. The new X axis parallel to the baseline: $\mathbf{r}_1 = (\mathbf{c}_1 - \mathbf{c}_2)/||\mathbf{c}_1 - \mathbf{c}_2||$
2. The new Y axis orthogonal to X (mandatory) and to $\mathbf{k}$: $\mathbf{r}_2 = \mathbf{k} \wedge \mathbf{r}_1$
3. The new Z axis orthogonal to XY (mandatory) : $\mathbf{r}_3 = \mathbf{r}_1 \wedge \mathbf{r}_2$

In point 2, $\mathbf{k}$ is an arbitrary unit vector, that fixes the position of the new Y axis in the plane orthogonal to X. We take it equal to the Z unit vector of the old left matrix, thereby constraining the new Y axis to be orthogonal to both the new X and the old left Z.

We formalize analytically the rectification requirements in Appendix A, where we also show that the algorithm given in the present section satisfies those requirements.

4 The rectifying transformation

In order to rectify – let's say – the left image, we need to compute the transformation mapping the image plane of $\tilde{\mathbf{P}}_{o1} = [\mathbf{Q}_{o1}|\tilde{\mathbf{q}}_{o1}]$ onto the image plane of $\tilde{\mathbf{P}}_{n1} = [\mathbf{Q}_{n1}|\tilde{\mathbf{q}}_{n1}]$. We will see that the sought transformation is the collinearity given by the 3×3 matrix $\mathbf{T}_1 = \mathbf{Q}_{n1}\mathbf{Q}_{o1}^{-1}$. The same result applies to the right image.

For any 3-D point $\mathbf{w}$ we can write

$$\begin{cases} \tilde{\mathbf{m}}_{o1} = \tilde{\mathbf{P}}_{o1}\tilde{\mathbf{w}} \\ \tilde{\mathbf{m}}_{n1} = \tilde{\mathbf{P}}_{n1}\tilde{\mathbf{w}}. \end{cases} \quad (11)$$

According to (8) , the equations of the optical rays are the following (since rectification does not move the optical center):

$$\begin{cases} \mathbf{w} = \mathbf{c}_1 + \lambda_o \mathbf{Q}_{o1}^{-1}\tilde{\mathbf{m}}_{o1} \\ \mathbf{w} = \mathbf{c}_1 + \lambda_n \mathbf{Q}_{n1}^{-1}\tilde{\mathbf{m}}_{n1}; \end{cases} \quad (12)$$

hence

$$\tilde{\mathbf{m}}_{n1} = \lambda \mathbf{Q}_{n1}\mathbf{Q}_{o1}^{-1}\tilde{\mathbf{m}}_{o1}, \quad (13)$$

where λ is an arbitrary scale factor (it is an equality between homogeneous quantities).

The transformation $\mathbf{T}_1$ is then applied to the original left image to produce the rectified image, as in Figure 6. Note that the pixels (integer-coordinate positions) of the rectified image correspond, in general, to non-integer positions on the original image plane. Therefore, the gray levels of the rectified image are computed by bilinear interpolation.

5 Summary of the rectification algorithm

The rectification algorithm can be summarized as follows:

- Given a stereo pair of images `I1,I2` and PPMs `Po1,Po2` (obtained by calibration);
- compute `[T1,T2,Pn1,Pn2] = rectify(Po1,Po2)` (see MATLAB code);
- rectify images by applying `T1` and `T2`.

Reconstruction of 3-D points by triangulation (Hartley and Sturm, 1997) can be performed from the rectified images directly, using `Pn1,Pn2`.

Given the high diffusion of stereo in research and applications, we have endeavored to make our algorithm as easily reproducible and usable as possible. To this purpose, we give the working MATLAB code of the algorithm; the code is simple and compact (22 lines), and the comments enclosed make it understandable without knowledge of MATLAB.

```
function [T1,T2,Pn1,Pn2] = rectify(Po1,Po2)

% RECTIFY: compute rectification matrices

% factorize old PPMs
[A1,R1,t1] = art(Po1);
[A2,R2,t2] = art(Po1);

% optical centers (unchanged)
c1 = - inv(Po1(:,1:3))*Po1(:,4);
c2 = - inv(Po2(:,1:3))*Po2(:,4);

% new x axis (= direction of the baseline)
v1 = (c1-c2);
% new y axes (orthogonal to new x and old z)
v2 = extp(R1(3,:)',v1);
% new z axes (orthogonal to baseline and y)
v3 = extp(v1,v2);

% new extrinsic parameters
R = [v1'/norm(v1)
     v2'/norm(v2)
     v3'/norm(v3)];
% translation is left unchanged

% new intrinsic parameters (arbitrary)
A = (A1 + A2)./2;
A(1,2)=0; % no skew

% new projection matrices
Pn1 = A * [R -R*c1 ];
Pn2 = A * [R -R*c2 ];

% rectifying image transformation
T1 = Pn1(1:3,1:3)* inv(Po1(1:3,1:3));
T2 = Pn2(1:3,1:3)* inv(Po2(1:3,1:3));

------------------------

function [A,R,t] = art(P)
% ART: factorize a PPM as  P=A*[R;t]

Q = inv(P(1:3, 1:3));
[U,B] = qr(Q);

R = inv(U);
t = B*P(1:3,4);
A = inv(B);
A = A ./A(3,3);
```

A "rectification kit" including C and MATLAB implementation of the algorithm, data sets and documentation can be found on line [1].

6 Experimental results

We ran tests to verify that the algorithm performed rectification correctly, and also to check that the accuracy of the 3-D reconstruction did not decrease when performed from the rectified images directly.

6.1 Correctness

The tests used both synthetic and real data. Each set of synthetic data consisted of a cloud of 3-D points and a pair of PPMs. For reasons of space, we report only two examples. Figure 5 shows the original and rectified images with a nearly rectified stereo rig: the camera translation was $-[100\ 2\ 3]$ mm and the rotation angles roll=1.5^o, pitch=2^o, yaw=1^o. Figure 4 shows the same with a more general geometry: the camera translation was $-[100\ 20\ 30]$ mm and the rotation angles roll=19^o pitch=32^o and yaw=5^o.

Real-data experiments used calibrated stereo pairs, courtesy of INRIA-Syntim. We show the results obtained with a nearly rectified stereo rig (Figure 6) and with a more general stereo geometry (Figure 7). The right image of each pair shows three epipolar lines corresponding to the points marked by a cross in the left image. The pixel coordinates of the rectified images are not constrained to lie in any special part of the image plane, and an arbitrary translation were applied to both images to bring them in a suitable region of the plane; then the output images were cropped to the size of the input images. In the case of the "Sport" stereo pair (image size 768×576), we started from the following camera matrices:

[1] http://www.dimi.uniud.it/~fusiello/rect.html

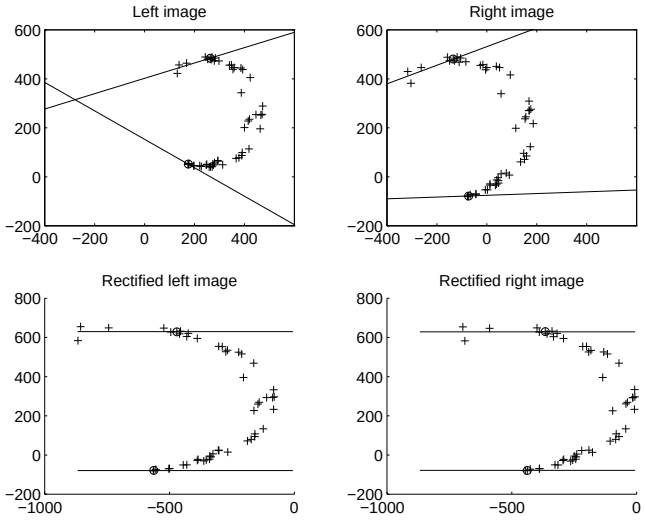


Fig. 4. General synthetic stereo pair (top) and rectified pair (bottom). The figure shows the epipolar lines of the points marked with a circle in both images.

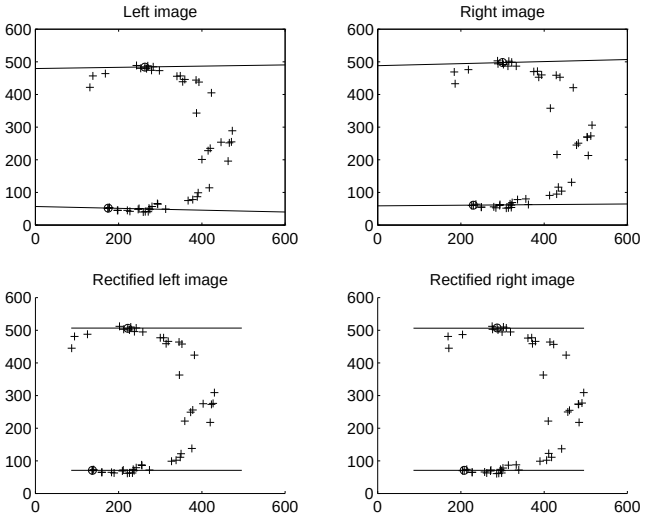


Fig. 5. Nearly rectified synthetic stereo pair (top) and rectified pair (bottom). The figure shows the epipolar lines of the points marked with a circle in both images.

$$\mathbf{P}_{o1} = \begin{bmatrix} 9.765e+2 & 5.382e+1 & -2.398e+2 & 3.875e+5 \\ 9.849e+1 & 9.333e+2 & 1.574e+2 & 2.428e+5 \\ 5.790e-1 & 1.108e-1 & 8.077e-1 & 1.118e+3 \end{bmatrix}$$

$$\mathbf{P}_{o2} = \begin{bmatrix} 9.767e+2 & 5.376e+1 & -2.400e+2 & 4.003e+4 \\ 9.868e+1 & 9.310e+2 & 1.567e+2 & 2.517e+5 \\ 5.766e-1 & 1.141e-1 & 8.089e-1 & 1.174e+3 \end{bmatrix}.$$

After adding the statement `A(1,3) = A(1,3) + 160` to the `rectify` program, to keep the rectified image in the center of the 768×576 window, we obtained the following rectified camera matrices:

$$\mathbf{P}_{n1} = \begin{bmatrix} 1.043e+3 & 7.452e+1 & -2.585e+2 & 4.124e+5 \\ 1.165e+2 & 9.338e+2 & 1.410e+2 & 2.388e+5 \\ 6.855e-1 & 1.139e-1 & 7.190e-1 & 1.102e+3 \end{bmatrix}$$

$$\mathbf{P}_{n2} = \begin{bmatrix} 1.043e+3 & 7.452e+1 & -2.585e+2 & 4.069e+4 \\ 1.165e+2 & 9.338e+2 & 1.410e+2 & 2.388e+5 \\ 6.855e-1 & 1.139e-1 & 7.190e-1 & 1.102e+3 \end{bmatrix}.$$

6.2 Accuracy

In order to evaluate the errors introduced by rectification on reconstruction, we compared the accuracy of 3-D reconstruction computed from original and rectified images. We used synthetic, noisy images of random clouds of 3-D points. Imaging errors were simulated by perturbing the image coordinates, and calibration errors by perturbing the intrinsic and extrinsic parameters, both with additive, Gaussian noise. Reconstruction were performed using the Linear-Eigen method, described in (Hartley and Sturm, 1997).

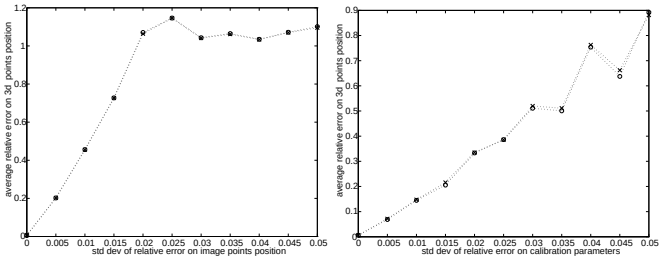


Fig. 8. Reconstruction error vs noise levels in the image coordinates (left) and calibration parameters (right) for the general synthetic stereo pair. Crosses refer to reconstruction from rectified images, circles to reconstruction from unrectified images.

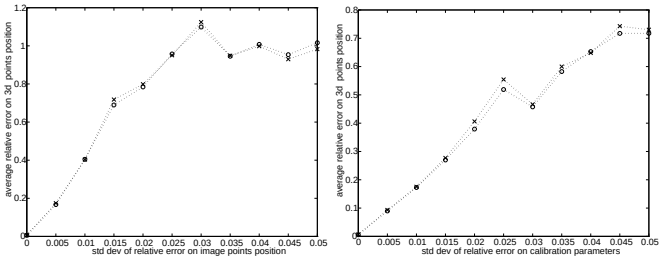


Fig. 9. Reconstruction error vs noise levels in the image coordinates (left) and calibration parameters (right) for the nearly rectified synthetic stereo pair. Crosses refer to reconstruction from rectified images, circles to reconstruction from unrectified images.

Figures 8 and 9 show the average (over the set of points) relative error measured on 3-D point position, plotted against noise. Figure 8 shows the results for the stereo rig used in Figure 4, and Figure 9 for the one used in Figure 5. Each point plotted is an average over 100 independent trials. The abscissa is the standard deviation of the relative error on coordinates of image point or calibration parameters.

7 Conclusion

Dense stereo matching is greatly simplified if images are rectified. We have developed a simple algorithm, easy to

Fig. 6. "Sport" stereo pair (top) and rectified pair (bottom). The right pictures plot the epipolar lines corresponding to the points marked in the left pictures.

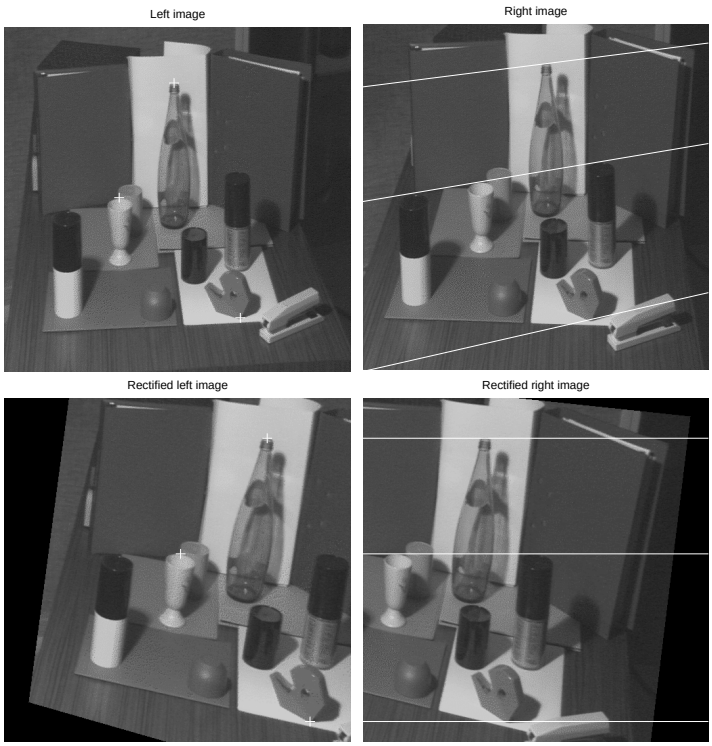


Fig. 7. "Color" stereo pair (top) and rectified pair (bottom). The right pictures plot the epipolar lines corresponding to the points marked in the left pictures.

understand and to use. Its correctness has been demonstrated analytically and by experiments. Our tests show that reconstructing from the rectified image does not introduces appreciable errors compared with reconstructing from the original images. We believe that a general rectification algorithm, together with the material we have made available on line, can prove a useful resource for the research and application communities alike.

Acknowledgement. This work benefited from discussions with Bruno Caprile, and was partially supported by grants from the British Council-MURST/CRUI and EPSRC (GR/L18716). Stereo pairs are courtesy of INRIA-Syntim (Copyright).

References

Ayache, N. and Lustman, F. (1991). Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:73–85.

Caprile, B. and Torre, V. (1990). Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4:127–140.

Dhond, U. R. and Aggarwal, J. K. (1989). Structure from stereo – a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510.

Faugeras, O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge,MA.

Fusiello, A., Roberto, V., and Trucco, E. (1997). Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–863, Puerto Rico. IEEE Computer Society Press.

Hartley, R. and Gupta, R. (1993). Computing matched-epipolar projections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 549–555.

Hartley, R. I. (1998). Theory and practice of projective rectification. *International Journal of Computer Vision*. To appear.

Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157.

Papadimitriou, D. V. and Dennis, T. J. (1996). Epipolar line estimation and rectification for stereo images pairs. *IEEE Transactions on Image Processing*, 3(4):672–676.

Robert, L. (1996). Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325.

Robert, L., Zeller, C., Faugeras, O., and Hébert, M. (1997). Applications of non-metric vision to some visually-guided robotics tasks. In Aloimonos, Y., editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 5, pages 89–134. Lawrence Erlbaum Associates.

A Analytical results

In this section we (i) formulate analytically the rectification requirements, and (ii) prove that our algorithm yields two PPMs $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ that satisfies such requirements. In order to improve readability, the transpose sign in scalar products will be omitted. All vector products are scalar products, unless otherwise noted.

Definition 1. *A pair of PPMs $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ are said to be rectified if, for any point $\mathbf{m}_1 = (u_1, v_1)^\top$ in the left image, the equation of its epipolar line in the right image is $v_2 = v_1$, and, for any point $\mathbf{m}_2 = (u_2, v_2)^\top$ in the right image, the equation of its epipolar line in the left image is $v_1 = v_2$.*

In the following, we shall write:

$$\tilde{\mathbf{P}}_{n1} = \left[\begin{array}{c|c} \mathbf{s}_1^\top & s_{14} \\ \mathbf{s}_2^\top & s_{24} \\ \mathbf{s}_3^\top & s_{34} \end{array}\right] = [\mathbf{S}|\tilde{\mathbf{s}}] \quad \tilde{\mathbf{P}}_{n2} = \left[\begin{array}{c|c} \mathbf{d}_1^\top & d_{14} \\ \mathbf{d}_2^\top & d_{24} \\ \mathbf{d}_3^\top & d_{34} \end{array}\right] = [\mathbf{D}|\tilde{\mathbf{d}}]. \tag{14}$$

Proposition 1. *Two perspective projection matrices $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ are rectified if and only if*

$$\left\{\begin{array}{l} \mathbf{s}_1\mathbf{c}_2 + s_{14} \neq 0 \\ \mathbf{s}_2\mathbf{c}_2 + s_{24} = 0 \\ \mathbf{s}_3\mathbf{c}_2 + s_{34} = 0 \end{array}\right. \text{and} \left\{\begin{array}{l} \mathbf{d}_1\mathbf{c}_1 + d_{14} \neq 0 \\ \mathbf{d}_2\mathbf{c}_1 + d_{24} = 0 \\ \mathbf{d}_3\mathbf{c}_1 + d_{34} = 0 \end{array}\right. \tag{15}$$

and

$$\frac{\mathbf{s}_2\mathbf{w} + s_{24}}{\mathbf{s}_3\mathbf{w} + s_{34}} = \frac{\mathbf{d}_2\mathbf{w} + d_{24}}{\mathbf{d}_3\mathbf{w} + d_{34}}, \tag{16}$$

where $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ are written as in (14) and $\mathbf{c}_1$ and $\mathbf{c}_2$ are the respective optical center's coordinates.

Proof. As we know, the epipolar line of $\tilde{\mathbf{m}}_2$ is the projection of its optical ray onto the left camera, hence its parametric equation writes:

$$\tilde{\mathbf{m}}_1 = \tilde{\mathbf{P}}_{n1}\begin{bmatrix}\mathbf{c}_2 \\ 1\end{bmatrix} + \tilde{\mathbf{P}}_{n1}\begin{bmatrix}\lambda\mathbf{D}^{-1}\tilde{\mathbf{m}}_2 \\ 0\end{bmatrix} = \tilde{\mathbf{e}}_1 + \lambda\mathbf{S}\mathbf{D}^{-1}\tilde{\mathbf{m}}_2 \tag{17}$$

where $\tilde{\mathbf{e}}_1$, the epipole, is the projection of the conjugate optical center $\mathbf{c}_2$:

$$\tilde{\mathbf{e}}_1 = \tilde{\mathbf{P}}_{n1}\begin{bmatrix}\mathbf{c}_2 \\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{s}_1\mathbf{c}_2 + s_{14} \\ \mathbf{s}_2\mathbf{c}_2 + s_{24} \\ \mathbf{s}_3\mathbf{c}_2 + s_{34}\end{bmatrix}. \tag{18}$$

The parametric equation of the epipolar line of $\tilde{\mathbf{m}}_2$ in image coordinates becomes:

$$\left\{\begin{array}{l} u = [\mathbf{m}_1]_1 = \dfrac{[\tilde{\mathbf{e}}_1]_1 + \lambda[\tilde{\mathbf{n}}]_1}{[\tilde{\mathbf{e}}1]_3 + \lambda[\tilde{\mathbf{n}}]_3} \\ \\ v = [\mathbf{m}_1]_2 = \dfrac{[\tilde{\mathbf{e}}_1]_2 + \lambda[\tilde{\mathbf{n}}]_2}{[\tilde{\mathbf{e}}_1]_3 + \lambda[\tilde{\mathbf{n}}]_3} \end{array}\right. \tag{19}$$

where $\tilde{\mathbf{n}} = \mathbf{S}\mathbf{D}^{-1}\tilde{\mathbf{m}}_2$, and $[.]_i$ is the projection operator extracting the ith component from a vector.

Analytically, the direction of each epipolar line can be obtained by taking the derivative of the parametric equations (19) with respect to λ:

$$\left\{\begin{array}{l} \dfrac{du}{d\lambda} = \dfrac{[\tilde{\mathbf{n}}]_1[\tilde{\mathbf{e}}_1]_3 - [\tilde{\mathbf{n}}]_3[\tilde{\mathbf{e}}_1]_1}{([\tilde{\mathbf{e}}_1]_3 + \lambda[\tilde{\mathbf{n}}]_3)^2} \\ \\ \dfrac{dv}{d\lambda} = \dfrac{[\tilde{\mathbf{n}}]_2[\tilde{\mathbf{e}}_1]_3 - [\tilde{\mathbf{n}}]_3[\tilde{\mathbf{e}}_1]_2}{([\tilde{\mathbf{e}}_1]_3 + \lambda[\tilde{\mathbf{n}}]_3)^2} \end{array}\right. . \tag{20}$$

Note that the denominator is the same in both components, hence it does not affect the direction of the vector. The epipole is rejected to infinity when $[\tilde{\mathbf{e}}_1]_3 = 0$. In this case, the direction of the epipolar lines in the right image doesn't depend on $\mathbf{n}$ and all the epipolar lines becomes parallel to vector $[[\tilde{\mathbf{e}}_1]_1\ [\tilde{\mathbf{e}}_1]_2]^\top$. The same holds, *mutatis mutandis*, for the left image.

Hence, epipolar lines are horizontal if and only if (15) holds. The vertical coordinate of conjugate points is the same in both image if and only if (16) holds, as can easily seen by plugging (14) into (5). □

Proposition 2. *The two camera matrices* $\tilde{\mathbf{P}}_{n1}$ *and* $\tilde{\mathbf{P}}_{n2}$ *produced by the rectification algorithm are rectified.*

Proof. We shall prove that, if $\tilde{\mathbf{P}}_{n1}$ and $\tilde{\mathbf{P}}_{n2}$ are built according to the rectification algorithm, then (15) and (16) hold.

From (9) we obtain

$$\begin{array}{lll} s_{14} = -\mathbf{s}_1\mathbf{c}_1 & d_{14} = -\mathbf{d}_1\mathbf{c}_2 & \mathbf{s}_1 = \mathbf{d}_1 \\ s_{24} = -\mathbf{s}_2\mathbf{c}_1 & d_{24} = -\mathbf{d}_2\mathbf{c}_2 & \mathbf{s}_2 = \mathbf{d}_2 \\ s_{34} = -\mathbf{s}_3\mathbf{c}_1 & d_{34} = -\mathbf{d}_3\mathbf{c}_2 & \mathbf{s}_3 = \mathbf{d}_3 \end{array} \tag{21}$$

From the factorization (2), assuming $\gamma = 0$, we obtain

$$\begin{bmatrix} \mathbf{s}_1^\top \\ \mathbf{s}_2^\top \\ \mathbf{s}_3^\top \end{bmatrix} = \mathbf{AR} = \begin{bmatrix} \alpha_u\mathbf{r}_1^\top + u_0\mathbf{r}_3^\top \\ \alpha_v\mathbf{r}_2^\top + v_0\mathbf{r}_3^\top \\ \mathbf{r}_3^\top \end{bmatrix} \tag{22}$$

From the construction of $\mathbf{R}$, we have that $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are mutually orthogonal, and $\mathbf{r}_1 = (\mathbf{c}_1 - \mathbf{c}_2)/\beta$ with $\beta = \|\mathbf{c}_1 - \mathbf{c}_2\|$.

From all these facts, the following four identity are derived:

$$\begin{aligned} \mathbf{s}_1(\mathbf{c}_1 - \mathbf{c}_2) &= \beta\mathbf{s}_1\mathbf{r}_1 = \beta(\alpha_u\mathbf{r}_1 + u_0\mathbf{r}_3)\mathbf{r}_1 = \\ &= \beta(\alpha_u\mathbf{r}_1\mathbf{r}_1 + u_0\mathbf{r}_3\mathbf{r}_1) = \beta\alpha_u \neq 0 \qquad (23) \\ \mathbf{s}_2(\mathbf{c}_1 - \mathbf{c}_2) &= \beta\mathbf{s}_2\mathbf{r}_1 = \beta(\alpha_v\mathbf{r}_2 + v_0\mathbf{r}_3)\mathbf{r}_1 = \\ &= \beta(\alpha_v\mathbf{r}_2\mathbf{r}_1 + v_0\mathbf{r}_3\mathbf{r}_1) = 0 \qquad (24) \\ \mathbf{s}_3(\mathbf{c}_1 - \mathbf{c}_2) &= \beta\mathbf{s}_3\mathbf{r}_1 = \beta\mathbf{r}_3\mathbf{r}_1 = 0 \qquad (25) \\ \mathbf{s}_2 \wedge \mathbf{s}_3 &= \mathbf{s}_2 \wedge \mathbf{r}_3 = \lambda(\mathbf{r}_2 \wedge \mathbf{r}_3) = \lambda\mathbf{r}_1 \qquad (26) \end{aligned}$$

The parameter λ in (26) is scalar taking into account that $\mathbf{s}_2$ is a linear combination of $\mathbf{r}_2$ and $\mathbf{r}_3$. Equation (15) follows easily from (23) (24)(25).

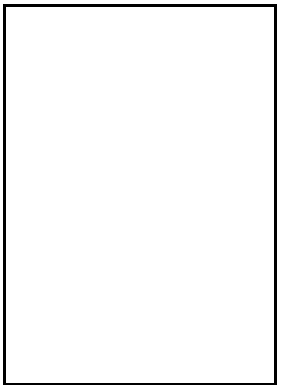
Equation (16) is equivalent to

$$(\mathbf{s}_2\mathbf{w} + s_{24})(\mathbf{d}_3\mathbf{w} + d_{34}) = (\mathbf{s}_3\mathbf{w} + s_{34})(\mathbf{d}_2\mathbf{w} + d_{24}). \tag{27}$$

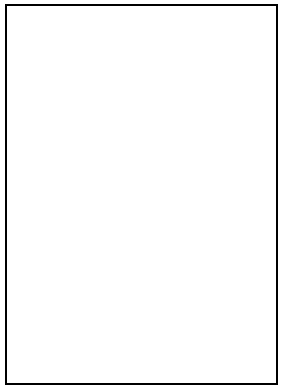
Expanding, and using (24),(26) and properties of the external product we obtain

$$\begin{aligned} -\mathbf{s}_2(\mathbf{c}_1 - \mathbf{c}_2)\mathbf{s}_3\mathbf{w} + (\mathbf{s}_2\mathbf{c}_1)(\mathbf{s}_3\mathbf{c}_2) - (\mathbf{s}_2\mathbf{c}_2)(\mathbf{s}_3\mathbf{c}_1) = \\ (\mathbf{s}_2\mathbf{c}_1)(\mathbf{s}_3\mathbf{c}_2) - (\mathbf{s}_2\mathbf{c}_2)(\mathbf{s}_3\mathbf{c}_1) = (\mathbf{s}_2 \wedge \mathbf{s}_3)(\mathbf{c}_1 \wedge \mathbf{c}_2) = \\ \lambda\mathbf{r}_1(\mathbf{c}_1 \wedge \mathbf{c}_2) = \lambda\beta(\mathbf{c}_1 - \mathbf{c}_2)(\mathbf{c}_1 \wedge \mathbf{c}_2) = 0. \end{aligned} \tag{28}$$

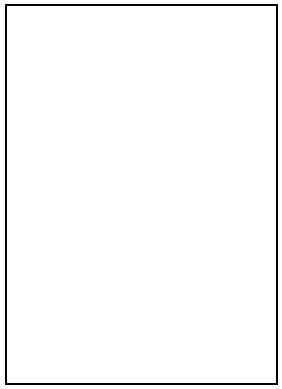
□



Andrea Fusiello received his Laurea (M.Sc.) degree in Computer Science from the Università di Udine, Italy in 1994 and his Ph.D in Computer Engineering from the Università di Trieste, Italy in 1999. In 1993-94 he worked within the Computer Vision Group at IRST, Italy. As a Research Fellow, he is now with the Dipartimento di Matematica e Informatica, Univesità di Udine. He has published papers on real-time task scheduling, autonomous navigation, stereo, and feature tracking. His present research is focused on 3-D computer vision, with applications to underwater robotics. He is a member of The International Association for Pattern Recognition (IAPR).



Manuel Trucco received his Laurea cum laude (MSc) in 1984 and the Research Doctorate (PhD) degree in 1990 from the University of Genoa, Italy, both in Electronic Engineering. Dr. Trucco has been active in machine vision research since 1984, at the EURATOM Joint Research Centre of the Commission of the European Communities (Ispra, Italy), and the Universities of Genoa and Edinburgh. currently a Lecturer in the Department of Computing and Electrical Engineering of Heriot-Watt University. His current research interests are in 3-D machine vision and its applications, particularly to underwater robotics. Dr Trucco has published more than 50 refereed papers and a book on 3-D vision. He is a member of IEEE, the British Machine Vision Association (BMVA), AISB, and a committee member of the British Machine Vision Association (Scottish Chapter). Further information can be found at http://www.cee.hw.ac.uk/ mtc/mtc.html



Alessandro Verri received the Laurea and Ph.D. in theoretical physics from the University of Genova in 1984 and 1988. Since 1989, he has been Ricercatore at the University of Genova (first at the Physics Department and, since the fall of 1997, at the Department of Computer and Information Science). He has published approximately 50 papers on various aspects of visual computation in man and machines – including stereo, motion analysis, shape representation, and object recognition – and coauthored a text-book on computer vision with Dr. E. Trucco. He has been visiting fellow at MIT, International Computer Science Institute, INRIA/IRISA (Rennes) and Heriot-Watt University. His current interests range from the study of computational problems of pattern recognition and computer vision to the development of visual systems for industrial inspection and quality control.

This article was processed by the author using the LaTeX style file from Springer-Verlag.