

A compact algorithm for rectification of stereo pairs

Andrea Fusiello¹, Emanuele Trucco², Alessandro Verri³

¹ Dipartimento Scientifico e Tecnologico, Università di Verona, Ca' Vignal 2, Strada Le Grazie, 37134 Verona, Italy; e-mail: fusiello@sci.univr.it

² Heriot-Watt University, Department of Computing and Electrical Engineering, Edinburgh, UK

³ INFN, Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Genova, Italy

Received: 25 February 1999 / Accepted: 2 March 2000

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. We report tests proving the correct behavior of our method, as well as the negligible decrease of the accuracy of 3D reconstruction performed from the rectified images directly.

Key words: Rectification – Stereo – Epipolar geometry

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) is made simpler, because search is done along the horizontal lines of the rectified images.

We assume that the stereo rig is *calibrated*, i.e., the cameras' internal parameters, mutual position and orientation are known. This assumption is not strictly necessary, but leads to a simpler technique. On the other hand, when reconstructing 3D 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 and Lustman (1991) introduced a rectification algorithm, in which a matrix satisfying a number of constraints is handcrafted. The distinction between necessary and arbitrary constraints is unclear. Some authors report rectification under restrictive assumptions; for

instance, Papadimitriou and Dennis (1996) assume a very restrictive geometry (parallel vertical axes of the camera reference frames). Recently, Hartley and Gupta (1993), Robert et al. (1997) and Hartley (1999) have introduced algorithms which perform rectification given a *weakly calibrated* stereo rig, i.e., a rig for which only points correspondences between images are given.

Latest work, published after the preparation of this manuscript includes Loop and Zhang (1999), Isgrò and Trucco (1999) and Pollefeys et al. (1999). Some of this work also concentrates on the issue of minimizing the rectified image distortion. We do not address this problem, partially because distortion is less severe than in the weakly calibrated case.

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. Moreover, given the shortage of easily reproducible, easily accessible and clearly stated algorithms, we have made the code available on the Web.

2 Camera model and epipolar geometry

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

2.1 Camera model

A pinhole camera is modeled by its *optical center* C and its *retinal plane* (or *image plane*) \mathcal{R} . A 3D point W is projected into an image point M given by the intersection of \mathcal{R} with the line containing C and W . The line containing C and orthogonal to \mathcal{R} is called the *optical axis* and its intersection with \mathcal{R} is the *principal point*. The distance between C and \mathcal{R} is the *focal length*.

Let $\mathbf{w} = [x \ y \ z]^T$ be the coordinates of W in the world reference frame (fixed arbitrarily) and $\mathbf{m} = [u \ v]^T$ the coordinates of M in the image plane (pixels). The mapping from 3D coordinates to 2D coordinates is the *perspec-*

tive 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 \mathbf{M} and \mathbf{W} , respectively; then, the perspective transformation is given by the matrix $\tilde{\mathbf{P}}$:

$$\tilde{\mathbf{m}} \cong \tilde{\mathbf{P}}\tilde{\mathbf{w}}, \quad (1)$$

where \cong means equal up to a 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

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

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

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

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-orthogonal $u - v$ axes.

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

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix} = [\mathbf{Q} \mid \tilde{\mathbf{q}}]. \quad (4)$$

In Cartesian coordinates, the projection (Eq. 1) writes

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

The *focal plane* is the plane parallel to the retinal plane that contains the optical center \mathbf{C} . The coordinates \mathbf{c} of \mathbf{C} are given by

$$\mathbf{c} = -\mathbf{Q}^{-1}\tilde{\mathbf{q}}. \quad (6)$$

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

$$\tilde{\mathbf{P}} = [\mathbf{Q} \mid -\mathbf{Q}\mathbf{c}]. \quad (7)$$

The *optical ray* associated to an image point \mathbf{M} is the line $\mathbf{M}\mathbf{C}$, i.e., the set of 3D points $\{\mathbf{w} : \tilde{\mathbf{m}} \cong \tilde{\mathbf{P}}\tilde{\mathbf{w}}\}$. In parametric form:

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

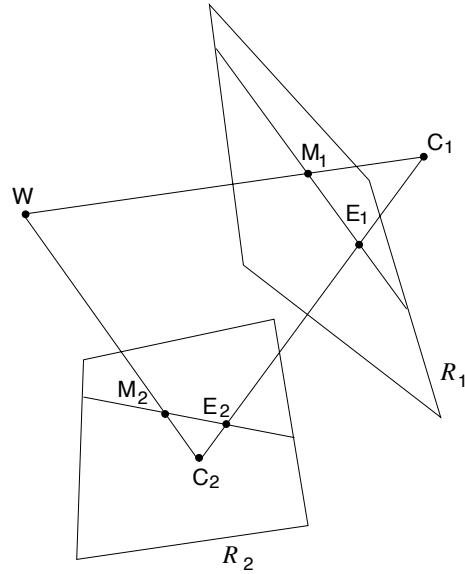


Fig. 1. Epipolar geometry. The epipole of the first camera \mathbf{E} is the projection of the optical center \mathbf{C}_2 of the second camera (and vice versa)

2.2 Epipolar geometry

Let us consider a stereo rig composed by two pinhole cameras (Fig. 1). Let \mathbf{C}_1 and \mathbf{C}_2 be the optical centers of the left and right cameras, respectively. A 3D point \mathbf{W} is projected onto both image planes, to points \mathbf{M}_1 and \mathbf{M}_2 , which constitute a conjugate pair. Given a point \mathbf{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 \mathbf{M}_1). Since \mathbf{M}_1 may be the projection of an arbitrary point on its optical ray, the epipolar line is the projection through \mathbf{C}_2 of the optical ray of \mathbf{M}_1 . All the epipolar lines in one image plane pass through a common point (\mathbf{E}_1 and \mathbf{E}_2 , respectively) called the *epipole*, which is the projection of the optical center of the other camera.

When \mathbf{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 $\mathbf{C}_1\mathbf{C}_2$ (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*.

3 Rectification of camera matrices

We assume that the stereo rig is *calibrated*, i.e., the 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 \mathbf{X} axis of both cameras. In addition, to have a proper rectification, conjugate points must have

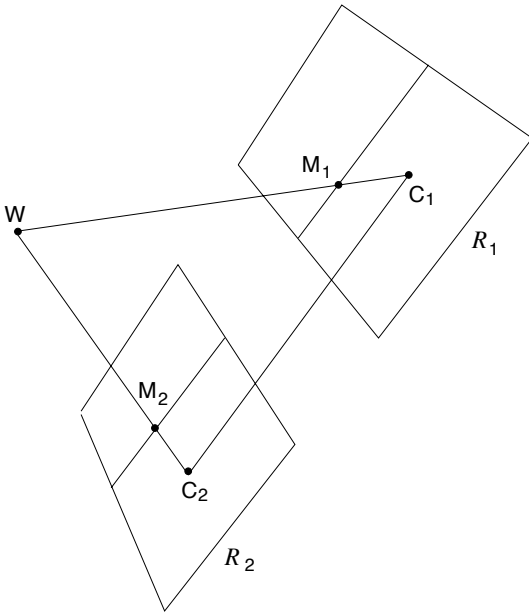


Fig. 2. Rectified cameras. Retinal planes are coplanar and parallel to the baseline

the *same vertical coordinate*. This is obtained by requiring that the new cameras have the same intrinsic parameters. Note that, being the focal length the same, retinal planes are coplanar too, as in Fig. 2.

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 cameras. 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 Eqs. 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 Eq. 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 .

This algorithm fails when the optical axis is parallel to the baseline, i.e., when there is a pure forward motion.

In Fusiello et al. (1998), we formalize analytically the rectification requirements, and we 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 $\tilde{\mathbf{P}}_{o1} = [\mathbf{Q}_{o1} \mid \tilde{\mathbf{q}}_{o1}]$ onto the image plane of $\tilde{\mathbf{P}}_{n1} = [\mathbf{Q}_{n1} \mid \tilde{\mathbf{q}}_{n1}]$. 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 3D point \mathbf{w} , we can write

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

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

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

hence,

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

The transformation \mathbf{T}_1 is then applied to the original left image to produce the rectified image, as in Fig. 5. 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.

Reconstruction of 3D points by triangulation (Hartley and Sturm, 1997) be performed from the rectified images directly, using $\mathbf{P}_{n1}, \mathbf{P}_{n2}$.

5 Summary of the rectification algorithm

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. The usage of the `rectify` function (see MATLAB code) is the following.

- Given a stereo pair of images $\mathbf{I}_1, \mathbf{I}_2$ and PPMs $\mathbf{P}_{o1}, \mathbf{P}_{o2}$ (obtained by calibration);
- compute $[\mathbf{T}_1, \mathbf{T}_2, \mathbf{P}_{n1}, \mathbf{P}_{n2}] = \text{rectify}(\mathbf{P}_{o1}, \mathbf{P}_{o2})$;
- rectify images by applying \mathbf{T}_1 and \mathbf{T}_2 .

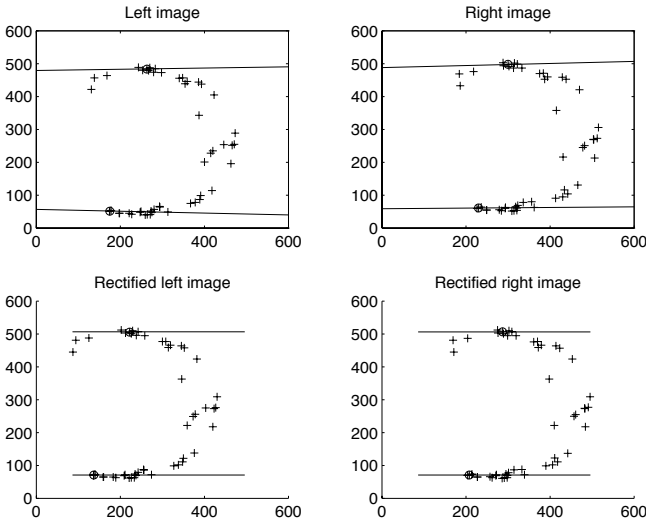


Fig. 3. Nearly rectified synthetic stereo pair (*top*) and rectified pair (*bottom*). The figure shows the epipolar lines of the points marked with a circle in both images

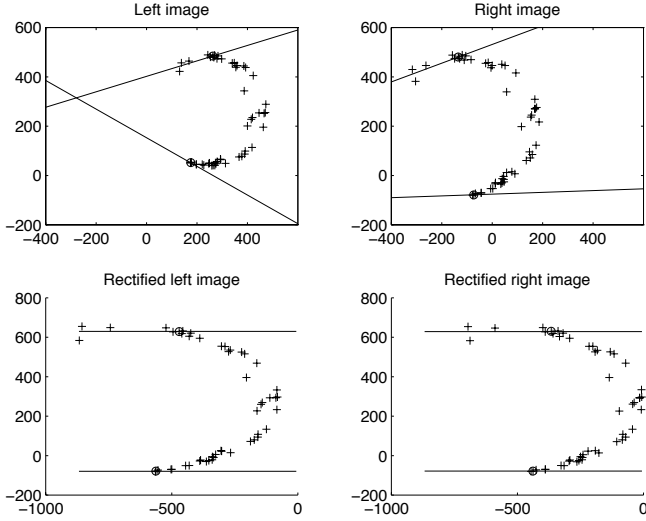


Fig. 4. General synthetic stereo pair (*top*) and rectified pair (*bottom*). The figure shows the epipolar lines of the points marked with a circle in both images

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

% RECTIFY: compute rectification matrices

% factorize old PPMs
[A1,R1,t1] = art(Po1);
[A2,R2,t2] = art(Po2);

% 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 = cross(R1(3,:),v1);
% new z axes (orthogonal to baseline and y)
v3 = cross(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 line¹.

6 Experimental results

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

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

Real-data experiments used calibrated stereo pairs, courtesy of INRIA-Syntim. We show the results obtained with a nearly rectified stereo rig (Fig. 5) and with a more general stereo geometry (Fig. 6). 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.sci.univr.it/~fusiello/rect.html>

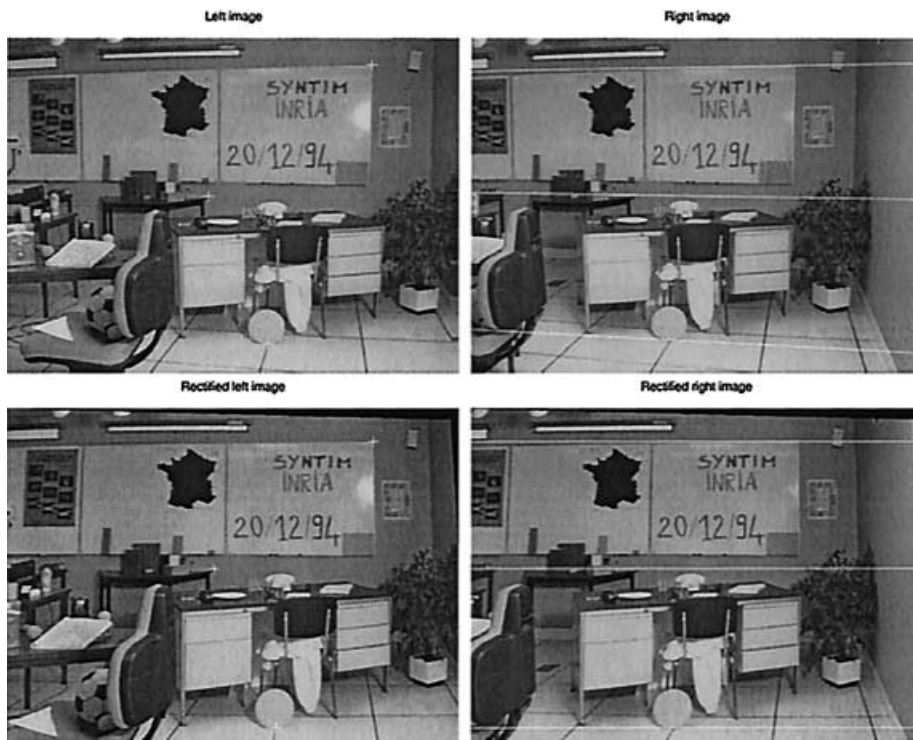


Fig. 5. “Sport” stereo pair (*top*) and rectified pair (*bottom*). The right pictures plot the epipolar lines corresponding to the points marked in the left pictures

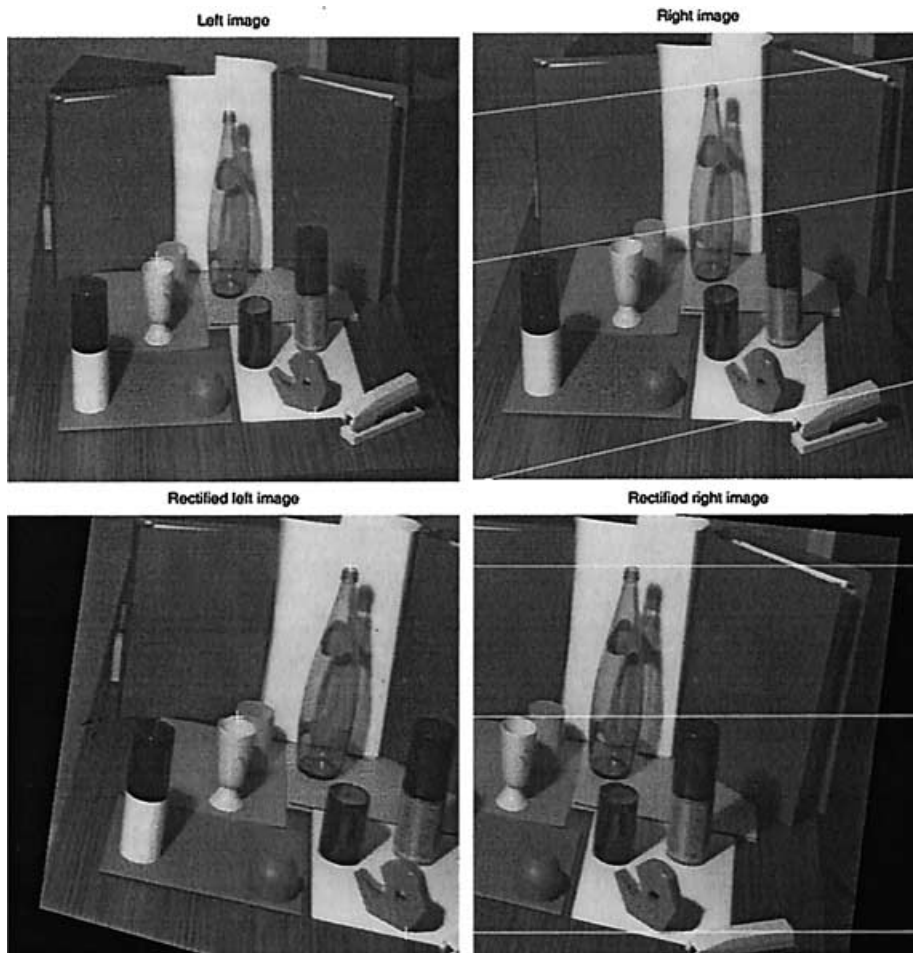


Fig. 6. “Color” stereo pair (*top*) and rectified pair (*bottom*). The right pictures plot the epipolar lines corresponding to the points marked in the left pictures

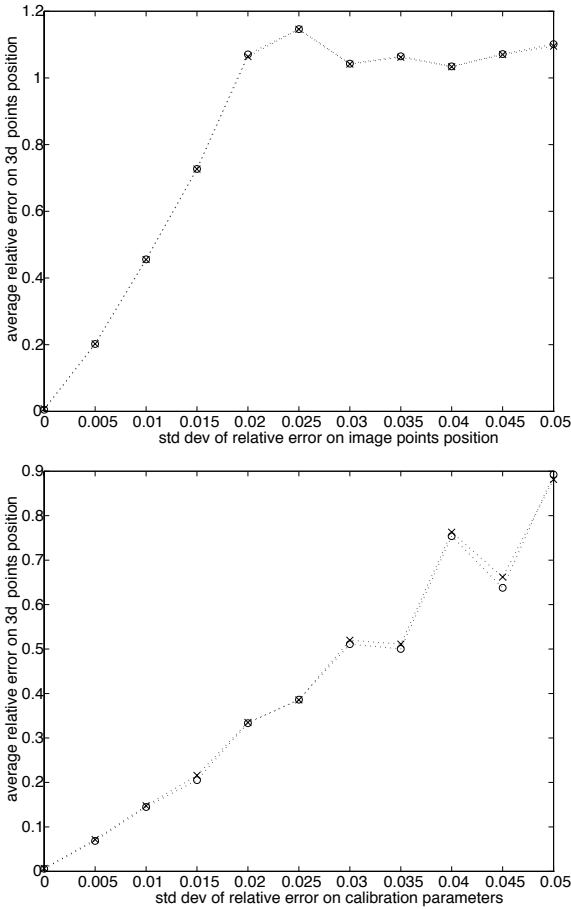


Fig. 7. Reconstruction error vs noise levels in the image coordinates (*left*) and calibration parameters (*right*) for the general synthetic stereo pair. *Crosses* refer to reconstruction from rectified images, *circles* to reconstruction from unrectified images

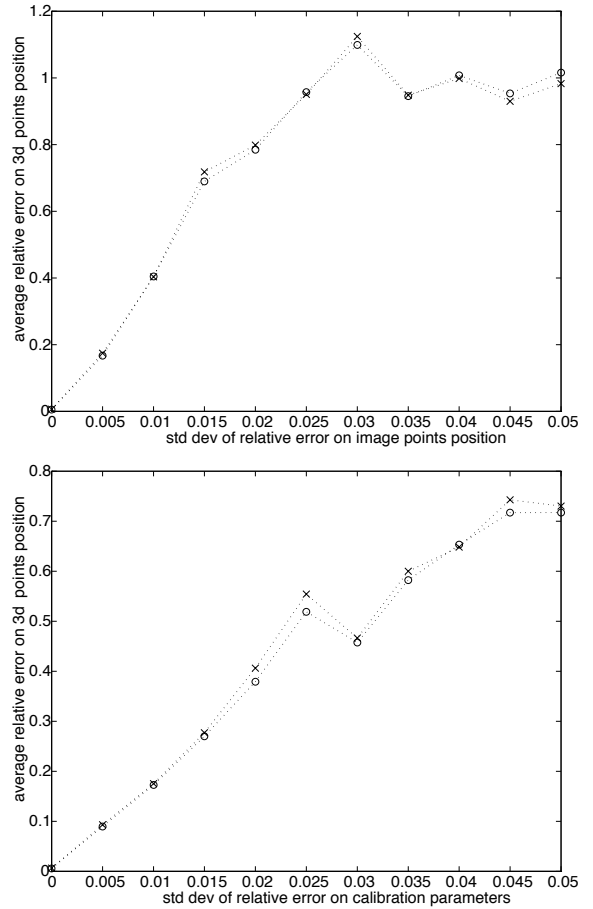


Fig. 8. Reconstruction error vs noise levels in the image coordinates (*left*) and calibration parameters (*right*) for the nearly rectified synthetic stereo pair. *Crosses* refer to reconstruction from rectified images, *circles* to reconstruction from unrectified images

$$\mathbf{P}_{o1} = \begin{bmatrix} 9.765 \cdot 10^2 & 5.382 \cdot 10^1 & -2.398 \cdot 10^2 & 3.875 \cdot 10^5 \\ 9.849 \cdot 10^1 & 9.333 \cdot 10^2 & 1.574 \cdot 10^2 & 2.428 \cdot 10^5 \\ 5.790 \cdot 10^{-1} & 1.108 \cdot 10^{-1} & 8.077 \cdot 10^{-1} & 1.118 \cdot 10^3 \end{bmatrix},$$

$$\mathbf{P}_{o2} = \begin{bmatrix} 9.767 \cdot 10^2 & 5.376 \cdot 10^1 & -2.400 \cdot 10^2 & 4.003 \cdot 10^4 \\ 9.868 \cdot 10^1 & 9.310 \cdot 10^2 & 1.567 \cdot 10^2 & 2.517 \cdot 10^5 \\ 5.766 \cdot 10^{-1} & 1.141 \cdot 10^{-1} & 8.089 \cdot 10^{-1} & 1.174 \cdot 10^3 \end{bmatrix}.$$

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

$$\mathbf{P}_{n1} = \begin{bmatrix} 1.043 \cdot 10^3 & 7.452 \cdot 10^1 & -2.585 \cdot 10^2 & 4.124 \cdot 10^5 \\ 1.165 \cdot 10^2 & 9.338 \cdot 10^2 & 1.410 \cdot 10^2 & 2.388 \cdot 10^5 \\ 6.855 \cdot 10^{-1} & 1.139 \cdot 10^{-1} & 7.190 \cdot 10^{-1} & 1.102 \cdot 10^3 \end{bmatrix},$$

$$\mathbf{P}_{n2} = \begin{bmatrix} 1.043 \cdot 10^3 & 7.452 \cdot 10^1 & -2.585 \cdot 10^2 & 4.069 \cdot 10^4 \\ 1.165 \cdot 10^2 & 9.338 \cdot 10^2 & 1.410 \cdot 10^2 & 2.388 \cdot 10^5 \\ 6.855 \cdot 10^{-1} & 1.139 \cdot 10^{-1} & 7.190 \cdot 10^{-1} & 1.102 \cdot 10^3 \end{bmatrix}.$$

Accuracy. In order to evaluate the errors introduced by rectification on reconstruction, we compared the accuracy of 3D reconstruction computed from original and rectified images. We used synthetic, noisy images of random clouds

of 3D 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 (Hartley and Sturm, 1997).

Figures 7 and 8 show the average (over the set of points) relative error measured on 3D point position, plotted against noise. Figure 7 shows the results for the stereo rig used in Fig. 4, and Fig. 8 for the one used in Fig. 3. 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 understand and to use. Its correctness has been demonstrated analytically and by experiments. Our tests show that reconstructing from the rectified image does not introduce 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.

Acknowledgements. Thanks to Bruno Caprile for valuable discussions and to the anonymous referees who made useful comments. This work 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, Lustman F (1991) Trinocular stereo vision for robotics. *IEEE Trans Pattern Anal Mach Intell* 13: 73–85
- Caprile B, Torre V (1990) Using vanishing points for camera calibration. *Int J Comput Vision* 4: 127–140
- Dhond UR, Aggarwal JK (1989) Structure from stereo – a review. *IEEE Trans Syst Man Cybern* 19(6): 1489–1510
- Faugeras O (1993) *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Mass.
- Fusiello A, Trucco E, Verri A (1998) Rectification with unconstrained stereo geometry. Research Memorandum RM/98/12, CEE Dept., Heriot-Watt University, Edinburgh, UK. <ftp://ftp.sci.univr.it/pub/Papers/Fusiello/RM-98-12.ps.gz>
- Hartley R (1999) Theory and practice of projective rectification. *Int J Comput Vision* 35(2): 1–16
- Hartley R, Gupta R (1993) Computing matched-epipolar projections. In: *CVPR93*, New York, NJ, pp 549–555
- Hartley RI, Sturm P (1997) Triangulation. *Comput Vision Image Understanding* 68(2): 146–157
- Isgro F, Trucco E (1999) Projective rectification without epipolar geometry. In: *CVPR99*, Fort Collins, CO, pp 1:94–99
- Loop C, Zhang Z (1999) Computing rectifying homographies for stereo vision. In: *CVPR99*, Fort Collins, CO, pp 1:125–131
- Papadimitriou DV, Dennis TJ (1996) Epipolar line estimation and rectification for stereo images pairs. *IEEE Trans Image Process* 3(4): 672–676
- Pollefeys M, Koch R, VanGool L (1999) A simple and efficient rectification method for general motion. In: *ICCV99*, Corfu, Greece, pp 496–501
- Robert L (1996) Camera calibration without feature extraction. *Comput Vision, Graphics Image Process* 63(2): 314–325
- Robert L, Zeller C, Faugeras O, Hébert M (1997) Applications of non-metric vision to some visually guided robotics tasks. In: Aloimonos Y (ed) *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Chap.5. Lawrence Erlbaum Assoc., pp 89–134



Andrea Fusiello received his Laurea (MSc) degree in Computer Science from the Università di Udine in 1994 and his PhD in Information Engineering from the Università di Trieste in 1999. He worked with the Computer Vision Group at IRST (Trento, IT) in 1993–94, and with the Machine Vision Laboratory at the Università di Udine from 1996 to 1998. He was a Visiting Research Fellow in the Department of Computing and Electrical Engineering of Heriot-

Watt University (UK) in 1998 and 1999. As a Research Associate, he is now with the DiST, Università di Verona. He has published papers on real-time task scheduling, autonomous vehicles navigation, stereo, and feature tracking. His present research is focused on 3D computer vision, with applications to underwater robotics. He is a member of the IAPR.



Emanuele Trucco received his Laurea cum laude (MSc) in 1984 and the Research Doctorate (PhD) degree in 1990 from the University of Genoa, IT, 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, IT), and the Universities of Genoa and Edinburgh. He is currently a Senior Lecturer in the Department of Computing and Electrical Engineering of Heriot-Watt University. His current research interests are in 3D machine vision and its

applications, particularly to underwater robotics. Dr. Trucco has published more than 80 refereed papers and a book on 3D 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) and of the IEE PG E4.



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 textbook on computer vision. He has been a 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.