Oriented Projective Geometry for Computer Vision

Stéphane Laveau Olivier Faugeras
e-mail: Stephane.Laveau@sophia.inria.fr,
Olivier.Faugeras@sophia.inria.fr

INRIA. 2004, route des Lucioles. B.P. 93. 06902 Sophia-Antipolis. FRANCE.

Abstract. We present an extension of the usual projective geometric framework for computer vision which can nicely take into account an information that was previously not used, i.e. the fact that the pixels in an image correspond to points which lie in front of the camera. This framework, called the oriented projective geometry, retains all the advantages of the unoriented projective geometry, namely its simplicity for expressing the viewing geometry of a system of cameras, while extending its adequation to model realistic situations.
We discuss the mathematical and practical issues raised by this new framework for a number of computer vision algorithms. We present different experiments where this new tool clearly helps.

1 Introduction

Projective geometry is now established as the correct and most convenient way to describe the geometry of systems of cameras and the geometry of the scene they record. The reason for this is that a pinhole camera, a very reasonable model for most cameras, is really a projective (in the sense of projective geometry) engine projecting (in the usual sense) the real world onto the retinal plane. Therefore we gain a lot in simplicity if we represent the real world as a part of a projective 3-D space and the retina as a part of a projective 2-D space.

But in using such a representation, we apparently loose information: we are used to think of the applications of computer vision as requiring a Euclidean space and this notion is lacking in the projective space. We are thus led to explore two interesting avenues. The first is the understanding of the relationship between the projective structure of, say, the environment and the usual affine and Euclidean structures, of what kind of measurements are possible within each of these three contexts and how can we use image measurements and/or a priori information to move from one structure to the next. This has been addressed in recent papers [7, 2]. The second is the exploration of the requirements of specific applications in terms of geometry. A typical question is, can this application be solved with projective information only, affine, or Euclidean. Answers to some of these questions for specific examples in robotics, image synthesis, and scene modelling are described in [11, 3, 4], respectively.

In this article we propose to add a significant feature to the projective framework, namely the possibility to take into account the fact that for a pinhole camera, both sides of the retinal plane are very different: one side corresponds to what is in front of the camera, one side to what is behind! The idea of visible points, i.e. of points located in front of the camera, is central in vision and the problem of enforcing the visibility of reconstructed points in stereo, motion or shape from X has not received a satisfactory answer as of today. A very interesting step in the direction of a possible solution has been taken by Hartley [6] with the idea of Cheirality invariants. We believe that our way of extending the framework of projective geometry goes significantly further.

Thus the key idea developed in this article is that even though a pinhole camera is indeed a projective engine, it is slightly more than that in the sense that we know for sure that all 3-D points whose images are recorded by the camera are in front of the camera. Hence the imaging process provides a way to tell apart both sides of the retinal plane. Our observation is that the mathematical framework for elaborating this idea already exists, it is the oriented projective geometry which has recently been proposed by Stolfi in his book [13].

2 A short introduction to oriented projective geometry

An n-dimensional projective space, $\mathcal{P}^n$ can be thought of as arising from an $n+1$ dimensional vector space in which we define the following relation between non zero vectors. To help guide the reader's intuition, it is useful to think of a non zero vector as defining a line through the origin. We say that two such vectors $\mathbf{x}$ and $\mathbf{y}$ are equivalent if and only if they define the same line. It is easily verified that this defines an equivalence relation on the vector space minus the zero vector. It is sometimes also useful to picture the projective space as the set of points of the unit sphere $\mathcal{S}^n$ of $\mathbb{R}^{n+1}$ with antipodal points identified. A point in that space is called a projective point; it is an equivalence class of vectors and can therefore be represented by any vector in the class. If $\mathbf{x}$ is such a vector, then $\lambda\mathbf{x}$, $\lambda \neq 0$ is also in the class and represents the same projective point.

In order to go from projective geometry to oriented projective geometry we only have to change the definition of the equivalence relation slightly:

$$\exists\lambda > 0 \text{ such that } \mathbf{y} = \lambda\mathbf{x} \tag{1}$$

where we now impose that the scalar λ be positive. The equivalence class of a vector now becomes the half-line defined by this vector. The set of equivalence classes is the oriented projective space $\mathcal{T}^n$ which can also be thought of as $\mathcal{S}^n$ but without the identification of antipodal points. A more useful representation, perhaps, is Stolfi's straight model [13] which describes $\mathcal{T}^n$ as two copies of $\mathbb{R}^n$, and an infinity point for every direction of $\mathbb{R}^{n+1}$, i.e. a sphere of points at infinity, each copy of $\mathbb{R}^n$ being the central projection of half of $\mathcal{S}^n$ onto the hyperplane of $\mathbb{R}^{n+1}$ of equation $x_1 = 1$. These two halves are referred to as the front range ($x_1 > 0$) and the back range ($x_1 < 0$) and we can think of the front half as the set of "real" points and the back half as the set of "phantom" points, or vice versa. Thus, given a point x of $\mathcal{T}^n$ of coordinate vector $\mathbf{x}$, the point represented by $-\mathbf{x}$ is different from x, it is called its antipode and noted $\neg x$.

The nice thing about $\mathcal{T}^n$ is that because it is homeomorphic to $\mathcal{S}^n$ (as opposed to $\mathcal{P}^n$ which is homeomorphic to $\mathcal{S}^n$ where the antipodal points have been identified), it is orientable. It is then possible to define a coherent orientation over the whole of $\mathcal{T}^n$: if we imagine moving a direct basis of the front range across the sphere at infinity into the back range and then back to the starting point of the front range, the final basis will have the same orientation as the initial one which is the definition of orientability. Note that this is not possible for $\mathcal{P}^n$ for even values of n.

3 Oriented projective geometry for computer vision

We now use the previous formalism to solve the problem of determining "front" from "back" for an arbitrary number of weakly calibrated pinhole cameras, i.e. cameras for which only image correspondences are known. We know that in this case only the projective structure of the scene can be recovered in general [1, 5]. We show that in fact a lot more can be recovered.

As usual, a camera is modeled as a linear mapping from $\mathcal{P}^3$ to $\mathcal{P}^2$ defined by a matrix $\mathbf{P}$ called the perspective projection matrix.The relation between a 3-D point M and its image m is $\mathbf{m} \simeq \mathbf{PM}$ where $\simeq$ denotes projective equality. By modeling the environment as $\mathcal{T}^3$, instead of $\mathcal{P}^3$, and by using the fact that the imaged points are in front of the retinal plane Π_f of the camera, we can orient that retinal plane in a natural way. In the case of two cameras, we can perform this orientation coherently and in fact extend it to the epipolar lines and the fundamental matrix. This applies also to any number of cameras. We call the process of orienting the focal plane of a camera orienting the camera.

3.1 Orienting the camera

The orienting of a camera is a relatively simple operation. It is enough to know the projective coordinates of a visible point. We say that this point is *in the front range* of the camera. By choosing one of the two points of $\mathcal{T}^3$ associated with this point of $\mathcal{P}^3$, we identify the front and the back range relatively to the focal plane of the camera.

The existence of such an information (the coordinates of a point in space and in the image) is verified in all practical cases. If the scene is a calibration grid, its space and image coordinates are known. In the case of *weak calibration*, a projective reconstruction is easily obtained by triangulation.

In order to define the orientation of the camera, we also need the assumption that the points we are considering do not project onto the line at infinity in the image plane. This is also verified in all practical cases, because no camera can see points on its focal plane.

Let us write the perspective projection matrix $\mathbf{P}$ as

$$\mathbf{P} = \begin{pmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_3^T \end{pmatrix}$$

where $\mathbf{l}_i$, $i = 1, 2, 3$ are 4×1 vectors defined up to scale. We know that the optical center is the point $\mathbf{C}$ verifying $\mathbf{PC} = 0$ and that $\mathbf{l}_3$ represents the focal plane Π_f of the camera. Π_f is a plane of $\mathcal{T}^3$ which we can orient by defining its positive side as being the front range of the camera and its negative side as being the back range of the camera. This is equivalent to choosing $\mathbf{l}_3$ such that, for example, the image of the points in front of the camera have their last coordinate positive, and negative for the points behind the camera. Again, this is just a convention, not a restriction.

The last coordinate of $\mathbf{m}$ is simply $\mathbf{l}_3.\mathbf{M}$. According to our conventions, this expression must be positive when $\mathbf{M}$ ($\mathbf{M} \in \mathcal{T}^3$) is *in front* of the focal plane. This determines the sign of $\mathbf{l}_3$ and consequently the sign of $\mathbf{P}$. $\mathbf{P}$ is then defined up to a *positive* scale factor.

Hence we have a clear example of the application of the oriented projective geometry framework: the retinal plane is a plane of $\mathcal{T}^3$ represented by two copies of $\mathbb{R}^2$ its front and back ranges in the terminology of section 1, which are two affine planes, and a circle of points at infinity. The front range "sees" the points in the front range of the camera, the back range "sees" the points in the back range of the camera.

The sign of $\mathbf{P}$ determines the orientation of the camera without ambiguity. A camera with an opposite orientation will look in the exact opposite direction, with the same projection characteristics. It is reassuring that these two different cameras are represented by two different mathematical objects. For clarity, in the Figure 1, consider that we are working in a plane containing $\mathbf{M}$ and $\mathbf{C}$. The scene is then $\mathcal{T}^2$ that we represent as a sphere, whereas the focal plane appears as a great circle. We know that the scene point will be between $\mathbf{C}$ and $\neg\mathbf{C}$.

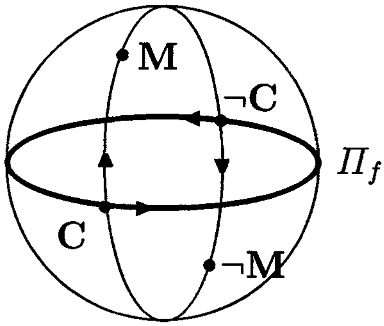

Fig. 1. A plane passing through $\mathbf{M}$ and $\mathbf{C}$ is represented as $\mathcal{S}^2$. The trace of the focal plane is an oriented line. It is oriented so that the front appears on the left when moving along the line.

3.2 Orienting the epipolar geometry

In this section, we consider a pair of cameras whose perspective projection matrices are $\mathbf{P}_1$ and $\mathbf{P}_2$. The epipoles are noted e_{12} and e_{21}.

The orienting of the cameras is not without having an incidence on the other attributes the stereo rig. The epipoles which are defined as the projections of the optical centers are oriented in the same fashion as the other points: if the optical center of the second camera is in front of (resp. behind) the focal plane of the first camera, the epipole has a positive (resp. negative) orientation, that is to say, a positive (resp. negative) last coordinate. This is achieved from the oriented pair of perspective projection matrices. The projective coordinates of the optical center $\mathbf{C}_2$ are computed from $\mathbf{P}_2$ by solving $\mathbf{P}_2\mathbf{C}_2 = 0$. From this information only we cannot decide which of the $\mathcal{T}^3$ objects corresponding to our projective points are the optical centers. Geometrically, this means that both $\mathbf{C}_2$ and $\neg\mathbf{C}_2$ are possible representations of the optical center. Of course, only one of them is correct. This is shown in Figure 2. If we choose the wrong optical center, the only incidence is that the orientation of all the epipoles and epipolar lines is going to be reversed.

The epipoles are then computed as images of the optical centers. In our case, $\mathbf{e} = \mathbf{P}\mathbf{C}'$. The epipolar lines can then be computed using $\mathbf{l}_{\mathbf{m}'} = \mathbf{e} \times \mathbf{m}$. As we can see, there are only two possible orientations for the set of epipolar lines. Setting the orientation of one of them imposes the orientation of all epipolar lines in the image. This ambiguity also goes away if we know the affine structure of the scene. The plane at infinity splits the sphere in two halves. We can choose its orientation so that M is on the positive side. Because C_1 and C_2 are *real* points, they also are on the positive side of the plane at infinity. This will discriminate between C_2 and $\neg C_2$.

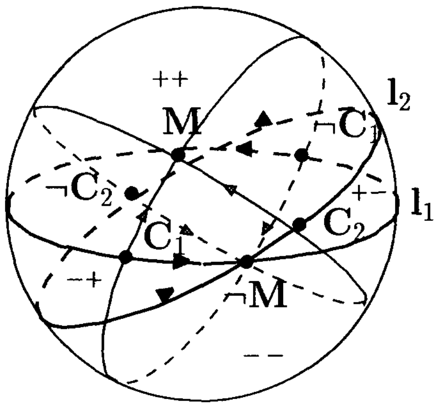

Fig. 2. The epipolar plane of M represented as the $\mathcal{T}^2$ sphere. l_1 and l_2 represent the intersection of the focal planes with the epipolar plane. The cameras are oriented so that M is in front of the cameras.

4 Applications

In this section, we show some applications of the oriented projective geometry in problems in computer vision involving weakly calibrated cameras.

4.1 Possible and impossible reconstructions in stereo

In this section, in order to keep the demonstrations clear and concise, we consider only two cameras. The extension to any given number of cameras is easy. It is a fact that for a stereo system, all reconstructed points must be in front of both cameras. This allows us to eliminate false matches which generate impossible reconstructions. This will be characterized by a scene crossing the focal plane of one or several cameras. The focal planes of the two cameras divide the space into four zones as shown in Figure 3. The reconstruction must lie in zone $++$ only.

When we reconstruct the points from the images, we obtain 3-D points in $\mathcal{P}^3$ which are pairs of points in $\mathcal{T}^3$. We have no way of deciding which of the antipodal points is a real point. Therefore, we choose the point in $\mathcal{T}^3$ to be in front of the first camera. We are then left with points possibly lying in $++$ and $+-$. The points in $+-$ are impossible points.

From this we see that we do not have a way to discriminate against points which are in the $--$ zone. These points can be real (i.e. in the front range of $\mathcal{T}^3$), but they will always have an antipodal point in $++$. On the other hand, points in $-+$ have their antipodal point in $+-$ and can always be removed.

We can eliminate the points in $--$ only if we know where is the front range of $\mathcal{T}^3$. This is equivalent to knowing the plane at infinty ant its orientation or equivalently the affine structure. Once we know the plane at infinity, we also know its orientation since

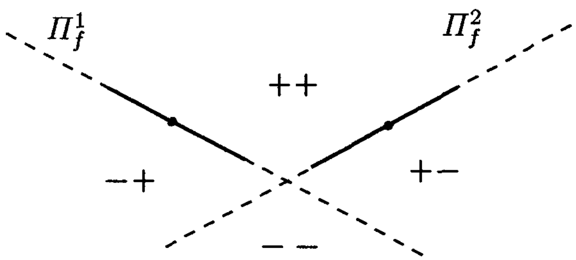

Fig. 3. Division of $\mathcal{T}^3$ in 4 zones.

our first point of reference used to orient the cameras must be *in front*. We can now constrain every reconstructed point to be in the front range of $\mathcal{T}^3$. This enables us to choose which point in $\mathcal{T}^3$ corresponds to the point in $\mathcal{P}^3$. The points appearing in $--$ can be removed, their antipodal point being an impossible reconstruction also.

We are not implying that we can detect this way all of the false matches, but only that this inexpensive step[1] can improve the results at very little additional expense. It should be used in conjunction with other outlier detection methods like [14] and [15].

The method is simple. From our correspondences, we compute a fundamental matrix as in [8] for example. From this fundamental matrix, we obtain two perspective projection matrices [1, 5], up to an unknown projective transformation. We then orient, perhaps arbitrarily, each of the two cameras. The reconstruction of the image points yields a cloud of pairs of points which can lie in the four zones.

In general one of the zones contains the majority of points because it corresponds to the real scene[2]. The points which are reconstructed in the other zones are then marked as incorrect and the cameras can be properly oriented so that the scene lies in front of them.

The pair of images in figure 4 has been taken with a conventional CCD camera. The correspondences were computed using correlation, then relaxation. An outlier rejection method was used to get rid of the matches which did not fulfill the epipolar constraints. Most outliers were detected using the techniques described in [14] and [15]. Still, these methods are unable to detect false matches which are consistent with the epipolar geometry. Using orientation, we discovered two other false matches which are marked as points 41 and 251. This is not a great improvement because most outliers have already been detected at previous steps, but these particular false matches could not have been detected using any other method.

4.2 Hidden surface removal

The problem is the following: given two points M^a and M^b in a scene which are both visible in image 1 but project to the same image point in image 2, we want to be able

[1] A projective reconstruction can be shown to be equivalent to a least-squares problem, that is to say a singular value decomposition.

[2] We are making the (usually) safe assumption that the correct matches outnumber the incorrect ones.

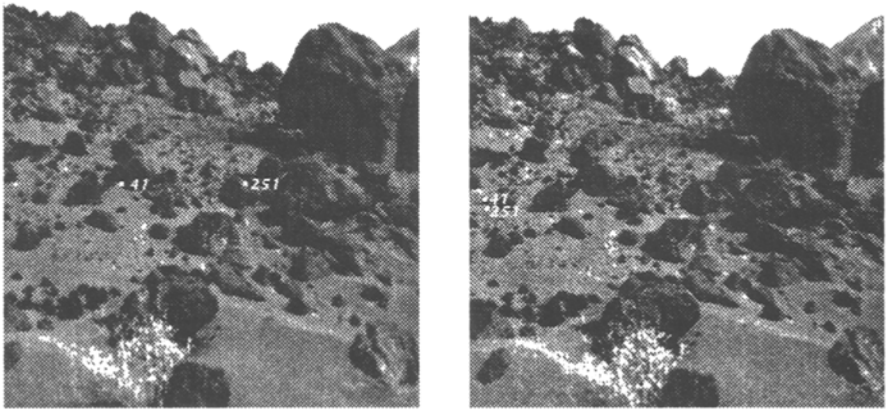

Fig. 4. Outliers detected with orientation only.

to decide from the two images only and their epipolar geometry which of the two scene points is actually visible in image 2 (see Figure 5). This problem is central in view transfer and image compression [3, 9].

It is not possible to identify the point using the epipolar geometry alone, because both points belong to the same epipolar line. We must identify the closest 3-D point to the optical center on the optical ray. It is of course the first object point when the ray is followed from the optical center towards infinity in the front range of the camera. It is a false assumption that it will necessarily be the closest point to the epipole in the image. In fact, the situation changes whenever the optical center of one camera crosses the focal plane of the other camera. This can be seen in the top part of Figure 5 where the closest point to the epipole switches from m_1^a to m_2^a when C_2 crosses Π_{f1} and becomes C_2' with the effect that e_{12} becomes e_{12}'.

We can use oriented projective geometry in order to solve this problem in a simple and elegant fashion. We have seen in the previous section that every point of the physical space projects onto the images with a sign describing its position with respect to the focal plane. We have also seen that the epipolar lines were oriented in a coherent fashion, namely from the epipole to the point. When the epipole is in the front range of the retinal plane as for e_{12} (right part of the bottom part of Figure 5), when we start from e_{12} and follow the orientation of the epipolar line, the first point we meet is m_1^a which is correct. When the epipole is in the back range of the retinal plane as for e_{12}' (left part of the bottom part of Figure 5), when we start from e_{12}' and follow the orientation of the epipolar line, we first go out to infinity and come back on the other side to meet m_1^a which is again correct!

Hence we have a way of detecting occlusion even if we use only projective information. The choice of which representant of C_2 we use will determine a possible orientation. But what happens when the chosen orientation is incorrect? In order to understand the problem better, we synthesized two views of the same object, using the same projection matrices, but with two different orientations. This is shown in Figure 6. The erroneous left view appears as seen "from the other side". The geometric interpretation

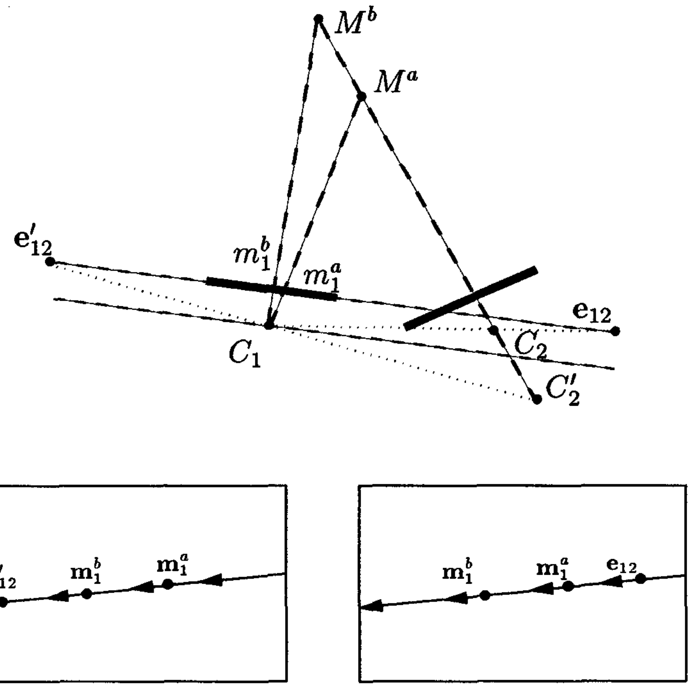

Fig. 5. Change of orientation when C_2 crosses the focal plane of the first camera.

is simple: The wrongly oriented camera looks in the opposite direction, but far enough to go through the sphere at infinity and come back to the other side of the object. Please note that without the use of oriented projective geometry, we would have a very large number of images, namely two possible orientations for each pixel.

4.3 Convex hulls

Another application of oriented projective geometry is the ability to build convex hulls of objects from at least two images. A method for computing the 3-D convex hull of an object from two views has been proposed before by Robert and Faugeras [12]. However, it is clear that the method fails if any point or optical center crosses any focal plane, as noted by the authors. Their approach is based on the homographies relating points in the images when their correspondents lie on a plane. Our approach makes full use of the oriented projective geometry framework to deal with the cases where their method fails. It consists in using once again the fact that the physical space is modelled as $\mathcal{T}^3$ and that in order to compute the convex hull of a set of 3-D points we only need to be able to compare the relative positions of these points with respect to any plane, i.e. to decide whether two points are or are not on the same side of a plane.

This is possible in the framework of oriented projective geometry. Let Π be a plane represented by the vector $\mathbf{u}$ which we do not assume to be oriented. Given two scene points represented by the vectors $\mathbf{M}$ and $\mathbf{M}'$ in the same zone (++ for example), comparing the signs of the dot products $\mathbf{u} \cdot \mathbf{M}$ and $\mathbf{u} \cdot \mathbf{M}'$ allows us to decide whether the

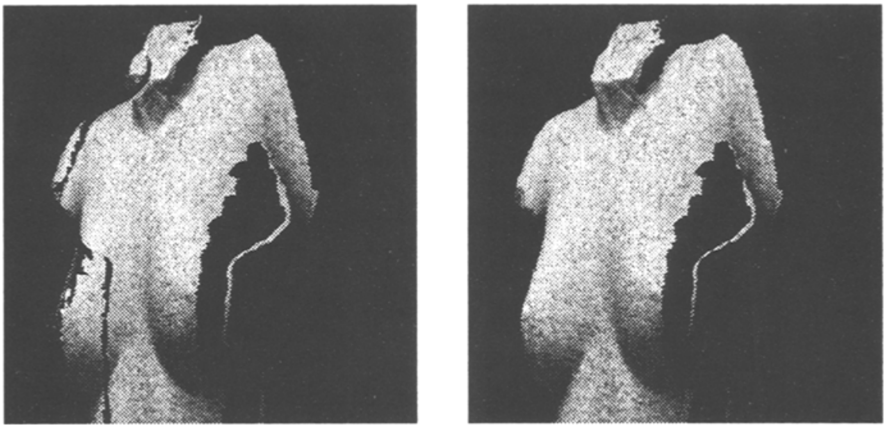

Fig. 6. Two synthesized images with cameras differing only in orientation. The image is incomplete because the source images did not cover the object entirely. The left image presents anomalies on the side due to the fact that the left breast of the mannequin is seen from the back. Hence, we see the back part of the breast first and there is a discontinuity line visible at the edge of the object in the source images. What we are seeing first are the last points on the ray. If the object was a head modelled completely, only the hair would be seen in one image, whereas the face would appear in the other. One image would appear seen from the back, and one from the front.

two points are on the same side of Π or not.

The usual algorithms for convex hull building can then be applied. There are of several sorts, a good reference being [10]. The results are of course identical to those of the method of Robert and Faugeras. The interested reader is then referred to [12] for details and results.

5 Conclusion

We have presented an extension of the usual projective geometric framework which can nicely take into account an information that was previously not used, i.e. the fact that we know that the pixels in an image correspond to points which lie in front of the camera. This framework, called the oriented projective geometry, retains all the advantages of the unoriented projective geometry, namely its simplicity for expressing the viewing geometry of a system of cameras, while extending its adequation to model realistic situations.

References

1. Olivier Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 563–578, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
2. Olivier Faugeras. Stratification of 3-d vision: projective, affine, and metric representations. *Journal of the Optical Society of America A*, 12(3):465–484, March 1995.
3. Olivier Faugeras and Stéphane Laveau. Representing three-dimensional data as a collection of images and fundamental matrices for image synthesis. In *Proceedings of the International Conference on Pattern Recognition*, pages 689–691, Jerusalem, Israel, October 1994. Computer Society Press.

4. Olivier Faugeras, Stéphane Laveau, Luc Robert, Cyril Zeller, and Gabriella Csurka. 3-d reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 145–168, Ascona, Switzerland, April 1995. ETH, Birkhauser Verlag. also INRIA Technical Report 2572.
5. Richard Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 761–764, Urbana Champaign, IL, June 1992. IEEE.
6. Richard I. Hartley. Cheirality invariants. In *Proceedings of the ARPA Image Understanding Workshop*, pages 745–753, Washington, DC, April 1993. Defense Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc.
7. Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. Technical Report UCB/CSD 93-772, Berkeley, 1993. Oct. 1993, revised July 1994.
8. Quang-Tuan Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques.* PhD thesis, Université de Paris-Sud, Centre d'Orsay, December 1992.
9. Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH*, Los Angeles, CA, August 1995.
10. F. Preparata and M. Shamos. *Computational Geometry.* Springer-Verlag, New-York, 1985.
11. L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually-guided robotics tasks. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter ? Lawrence Erlbaum Associates, 1996. to appear, also INRIA Technical Report 2584.
12. Luc Robert and Olivier Faugeras. Relative 3-D positioning and 3-D convex hull computation from a weakly calibrated stereo pair. *Image and Vision Computing*, 13(3):189–197, 1995. also INRIA Technical Report 2349.
13. Jorge Stolfi. *Oriented Projective Geometry, A Framework for Geometric Computations.* Academic Press, Inc., 1250 Sixth Avenue, San Diego, CA, 1991.
14. Philip Torr. *Motion Segmentation and Outlier Detection.* PhD thesis, Department of Engineering Science, University of Oxford, 1995.
15. Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 1994. to appear, also INRIA Research Report No.2273, May 1994.