ELSEVIER

Artificial Intelligence 78 (1995) 213-238

Artificial Intelligence

Understanding positioning from multiple images

Roger Mohr [1], Boubakeur Boufama [2], Pascal Brand [3]

Lifia-Inria, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France

Received October 1993; revised December 1994

Abstract

It is possible to recover the three-dimensional structure of a scene using only correspondences between images taken with uncalibrated cameras (faugeras 1992). The reconstruction obtained this way is only defined up to a projective transformation of the 3D space. However, this kind of structure allows some spatial reasoning such as finding a path. In order to perform more specific reasoning, or to perform work with a robot moving in Euclidean space, Euclidean or affine constraints have to be added to the camera observations. Such constraints arise from the knowledge of the scene: location of points, geometrical constraints on lines, etc. First, this paper presents a reconstruction method for the scene, then it discusses how the framework of projective geometry allows symbolic or numerical information about positions to be derived, and how knowledge about the scene can be used for computing symbolic or numerical relationships. Implementation issues and experimental results are discussed.

1. Introduction

1.1. Motivation

In the late seventies, Artificial Intelligence techniques were introduced into the field of image interpretation. Symbolic knowledge was introduced by means of geometric relations between image features. For instance, constraints like *above(sky, tree)* (see for instance [4]) reduce the labelling possibilities during the interpretation process. Expressing such a relation implicitly assumes that the camera is looking forward horizontally. More complex relationships like *inside* may also be used, however they are hard to define, as an image is only a 2D projection of the 3D scene.

[1] E-mail: mohr@imag.fr.

[2] Current address: Département d'Informatique, Université de Moncton, Moncton, New Brunswick, Canada E1A 3E9. E-mail: boufama@clement.info.unmoncton.ca.

[3] E-mail: brand@imag.fr.

SSDI 0004-3702(95)00035-6

The goal of this paper is to show how geometry provides a set of basic tools for numerically computing symbolic relationships, and to make explicit and computable what is often implicit in such relationships. The framework will point out the kind of geometric knowledge that is needed and how to explicitly compute relations like *above* and *inside*.

From a single image it is very hard to derive 3D relationships in the scene. An alternative is to use high level interpretation of the scene as is done in [7,29]. This framework allows high level reasoning, but on the other hand the knowledge required for this reasoning is very specific to the domain considered. Such approaches are complementary to the one presented here: we use very little *a priori* information and the outputs are mainly numerical 3D information and some simple symbolic relationships.

This paper presents a general way for expressing geometrical constraints and shows how to compute information such as 3D relationships between geometrical features observed in the images. Such numerical computations assume that the scene is observed by at least two cameras and the points are matched in the different images. However, we allow the camera position and orientation to be unknown. This allows us to avoid the burden of calibration. Since camera parameters vary when the focus and aperture change, the uncalibrated framework that we outline here can conveniently be placed in the field of active vision.

It was shown in [6,10] that reconstruction of the scene with uncalibrated cameras is possible. The reconstruction has no metric information and is defined up to a collineation of the 3D projective space. However, the use of this kind of reconstruction (i.e. non-metric information), has received more and more attention in the past few years. Projective space allows us to talk about collinearity, cross ratio, etc. However, relationships like *parallel_to, midpoint_of* are affine properties. Since affine space can be seen as a constrained projective space, additional knowledge needs to be provided. Euclidean space can be seen as a constrained affine space, so if Euclidean information is needed yet more knowledge must be provided (see [14] for an invariant based comparison and relationships between projective, affine and Euclidean geometries).

1.2. Outline

Projective, affine and Euclidean geometries form the basic framework of this paper. They are introduced in Section 2. Firstly we introduce the basic geometrical tools that are needed, and then, show how projective reconstructions can be obtained using the least squares approach. An outline of this approach was presented in [20]. Here, some new results on real images are discussed, illustrating both qualitative and quantitative aspects of the method.

Next, we consider different relationships between geometric entities. We will show that in fact the reconstruction done in projective space contains more than just projective information. In fact, so called quasi-affine information is available, from which it is possible to compute some qualitative affine relationships and convex hulls. However, affine information is not enough for some precise relationships. For instance, *above* relation needs additional information about the vertical direction.

This leads to a hierarchy of geometrical knowledge which allows us to define sets of symbolic relationships.

Classical 19th century geometry describes a hierarchy of embedded geometries: projective geometry, affine geometry, geometry of similarities and Euclidean geometry. However, other non-standard knowledge can be considered too. For instance the conservation of convexity leads to the notion of quasi-affine geometry, while preserving a particular direction like the vertical direction leads to a particular sub-affine geometry. From a practical point of view, this hierarchy of geometries corresponds to a hierarchy of constraints on the perceived scene. Section 3 shows how these constraints can be provided from *a priori* knowledge, and how they can be integrated into the general reconstruction framework.

Section 4 presents reconstruction results and examples of computations in the intermediate spaces. Finally, we discuss issues raised by the present work.

2. The geometrical framework

2.1. Introduction to projective geometry

This section aims at providing the reader with the geometrical background to understand the remainder of the paper.

We model the image formation system as a pure perspective projection, i.e. the camera model we adopt is the pinhole model. This is a good approximation of existing image acquisition systems. Sometimes this model has to be corrected for radial distortion [31], and if a very high accuracy is needed more sophisticated correction methods are necessary [17]. It has to be mentioned that these last models mainly correct the image in order to simulate the pure pinhole model.

Projective geometry deals elegantly with the general case of perspective projection and therefore provides clear understanding of the geometric aspects of image formation.

We will present a short introduction to the definitions and vocabulary of projective geometry. The reader is referred to [25] for a gentle introduction or to [15] for advanced vision oriented considerations on projective geometry.

Projective space

Consider the $(n+1)$-dimensional space $\mathbb{R}^{n+1} - \{(0,\ldots,0)\}$ with the equivalence relation:

$$(x_1,\ldots,x_{n+1}) \sim (x'_1,\ldots,x'_{n+1}) \quad \text{iff}$$
$$\exists \lambda \neq 0 \text{ such that } (x'_1,\ldots,x'_{n+1}) = \lambda(x_1,\ldots,x_{n+1}). \tag{1}$$

The quotient space obtained from this equivalence relation is the projective space $\mathcal{P}^n$. Thus the $(n+1)$-tuples of coordinates $(x_1,\ldots,x_{n+1})$ and $(x'_1,\ldots,x'_{n+1})$ represent the same point in the projective space and are called the homogeneous coordinates.

The usual n-dimensional affine space $\mathcal{A}^n$ is mapped into $\mathcal{P}^n$ through the correspondence Ψ:

$$\Psi : (x_1, \ldots, x_n) \rightarrow (x_1, \ldots, x_n, 1). \tag{2}$$

Note that Ψ is a one-to-one mapping and that only the points represented by $(x_1, \ldots, x_n, 0)$ are not reached. Ψ provides us with an understanding of the points $(x_1, \ldots, x_n, 1)$ which can be viewed as the usual point in the Euclidean space; it also provides us with an intuitive understanding of the remaining points: if we consider $(y_1, \ldots, y_n, 0)$ as the limit of $(y_1, \ldots, y_n, \lambda)$ while $\lambda \rightarrow 0$, i.e. the limit of $(y_1/\lambda, \ldots, y_n/\lambda, 1)$. This is the limit of a point in $\mathbb{R}^n$ going to the infinity in the direction $(y_1, \ldots, y_n)$. Therefore, we can consider $(y_1, \ldots, y_n, 0)$ as the "point at infinity" (or ideal point) in this direction.

A hyperplane H in $\mathcal{P}^n$ is defined by a homogeneous $(n+1)$-tuple of coefficients, $H = (a_1, \ldots, a_{n+1})$. This defines the set of points whose coordinates satisfy

$$\sum_{1}^{n+1} a_i x_i = HX^{\mathrm{T}} = 0.$$

A particular case is the ideal hyperplane $x_{n+1} = 0$: this is the hyperplane with all its points at infinity.

A **collineation** or projective transformation from $\mathcal{P}^n$ into $\mathcal{P}^n$ is the mapping defined by a full rank $(n+1) \times (n+1)$ matrix W such that the image of $(x_1, \ldots, x_{n+1})$ is defined in the usual way by:

$$W \begin{pmatrix} x_1 \\ \vdots \\ x_{n+1} \end{pmatrix} = \lambda \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_{n+1} \end{pmatrix}.$$

Since the column vectors are defined up to a scaling factor, the matrix W is too. Therefore, a collineation has $(n+1) \times (n+1) - 1$ degrees of freedom.

A **basis** of a projective space $\mathcal{P}^n$ is given by $n+2$ points, of which no $n+1$ lie in the same hyperplane. The canonical basis usually chosen is $(1,0,\ldots,0), \ldots, (0,0,\ldots,0,1)$ augmented with the "unit point" $(1,1,\ldots,1)$. For the regular 3D projective spaces, these four first points are the point at infinity on the x-axis, on the y-axis, on the z-axis and the origin, respectively.

A standard **affine transformation** maps easily into the projective transformation. In the general case an affine transformation in $\mathcal{A}^3$ is defined by a translation $t = (a, b, c)$, and a linear transformation (3×3 matrix) M on 3D vectors. The associated collineation is then defined by the matrix:

$$\begin{pmatrix} M & t^{\mathrm{T}} \\ 0\,0\,0 & 1 \end{pmatrix}.$$

Notice that in this case each point at infinity is mapped onto a point at infinity. Conversely, all collineations mapping points at infinity onto points at infinity are affine transformations.

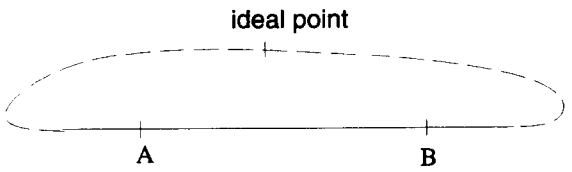

Fig. 1. Two points A and B on a projective line define two segments.

A projective transformation from $\mathcal{P}^n$ into $\mathcal{P}^k$ is defined by a $(k+1) \times (n+1)$ full rank matrix. A particular case is the perspective projection which maps the space $\mathcal{P}^3$ into the image plane $\mathcal{P}^2$.

It has to be noted that the addition of the hyperplane of ideal points changes the topology of the space. For instance, two points don't define a unique segment but define two segments: the first segment can be seen as the usual segment joining the two points in a standard affine space, the second one is the segment joining the two points too but passing through an ideal point. For this reason a projective line is often represented as a closed curve (see Fig. 1). Since the ideal points are unknown in projective space, the two segments linking two points are equivalent.

The basic projective invariant

The cross-ratio is the basic invariant in projective geometry: all other projective invariants can be derived from it [5].

Theorem . Let A, B, C, D be four collinear points, their *cross-ratio* is defined as:

$$[A,B;C,D] = \frac{\overline{AC}}{\overline{AD}} \times \frac{\overline{BD}}{\overline{BC}}; \tag{3}$$

this *cross-ratio* is invariant under any collineation.

The notation $\overline{AC}$ stands for the algebraic measure of the segment AC. This result was already established by the ancient Greek mathematicians. It can be extended to projective space using the following computational rules to deal with infinity:

$$\frac{\infty}{\infty} = 1, \qquad \frac{a}{\infty} = 0, \qquad \frac{\infty}{a} = \infty.$$

This theorem has an immediate application for locating a point on a line. Given 3 points, the position of the fourth is uniquely defined by the cross ratio of the four points. So, three points form a projective basis for the projective line.

2.2. *The reconstruction method*

Usually scene reconstruction requires a prior step: the calibration of the camera. Unfortunately, in many applications it is not possible to calibrate on-line, for example, if a calibration pattern is not available or if the camera is involved in visual tasks.

An alternative approach is to use points in the scene as a reference frame without knowing either their absolute coordinates or the camera parameters. This has been

investigated by several researchers in the past few years using projective geometry [18] or affine shape [27].

Under the hypothesis of parallel projection or orthography hypothesis, many other researchers [11, 12, 30] have developed similar methods for shape recovery, but these methods do not deal with the perspective projection case.

Using standard tools of projective geometry, Faugeras [6] developed a reconstruction method from perspective views. He demonstrates that it is possible to reconstruct three-dimensional scenes from point matches only, but such reconstruction can be defined only up to a collineation, i.e. a projective transformation.

Azarbayejani et al. [1] uses a recursive method to estimate motion and structure from tracked points. Their method gives good results and accounts for noise, however, they suppose that the focal length is constant between images and do not consider the case where the internal parameters are unknown and changing from one image to the next. Soatto et al. [26] propose another recursive method which can be compared to [1]. Their algorithm works with normalized image coordinates, and this is the main difference from the method proposed here.

Our approach has been inspired by the work of photogrammetrists. In particular, by the way they simultaneously calibrate a camera and reconstruct the scene, using accurately located beacons [2]. Szeliski and Kang's reconstruction method [28] can be considered close to our approach since both are based on global minimization. However, Szeliski and Kang suppose the knowledge of the internal parameters of the camera (they estimate these parameters in a prior step).

This section considers the problem of computing the location of points in the three-dimensional space given at least two perspective views taken with uncalibrated cameras.

We describe a reconstruction method that uses five points of the scene as a relative 3D frame. Unlike the linear method proposed by Faugeras [6] which is based on epipolar geometry, our method is direct in the sense that the reconstruction is obtained in one step using the minimum amount of information. Faugeras' method could be considered as an alternative for the initialization of our iterative algorithm, however we find our initial values with a different method explained in Section 4.

2.2.1. The basic equations

Consider v ($v \geqslant 2$) images of a scene composed of p points. For simplicity, assume that all the points have been matched in all the images, thus providing $p \times v$ image points. In fact, it is sufficient for a point to appear in at least 2 images.

Let $\{\boldsymbol{P}_i, i = 1, \ldots, p\}$ and $\{\boldsymbol{M}_j, j = 1, \ldots, v\}$ be the unknown 3D points and projection matrices, respectively.

For each image j, the point $\boldsymbol{P}_i$, represented by a column vector of its homogeneous coordinates $(x_i, y_i, z_i, t_i)^{\mathrm{T}}$ or its usual inhomogeneous coordinates $(X_i, Y_i, Z_i)^{\mathrm{T}} = (x_i/t_i, y_i/t_i, z_i/t_i)^{\mathrm{T}}$, is projected as the point $\boldsymbol{p}_{ij}$, represented by a column vector of its three homogeneous coordinates $(u_{ij}, v_{ij}, w_{ij})^{\mathrm{T}}$ or its usual inhomogeneous coordinates $(U_{ij}, V_{ij})^{\mathrm{T}}$.

In homogeneous coordinates, the perspective projection can be written as

$$\rho_{ij}\boldsymbol{p}_{ij} = \boldsymbol{M}_j\boldsymbol{P}_i, \qquad i = 1, \ldots, p, \quad j = 1, \ldots, v, \tag{4}$$

where ρ_{ij} is an unknown scaling factor.

Eq. (4) is usually written in the following way, hiding the scaling factor, and using the inhomogeneous coordinates of the image points:

$$\begin{cases} U_{ij} = \dfrac{m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}, \\ V_{ij} = \dfrac{m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}. \end{cases} \tag{5}$$

These equations express the collinearity of the space points ($\boldsymbol{P}_i$), and their corresponding projection points ($\boldsymbol{p}_{ij}$) with the centers of projection.

As we have p points and v images, this leads to $2 \times p \times v$ equations. There are $11 \times v$ unknowns for the $\boldsymbol{M}_j$ (which are defined up to a scaling factor), plus $3 \times p$ for the space points. If v and p are large enough the system (5) becomes overconstrained.

For a general configuration, this system has a solution [6], but this solution can only be defined up to a collineation. Eq. (4) can be written as

$$\rho_{ij}\boldsymbol{p}_{ij} = \boldsymbol{M}_j\boldsymbol{P}_i = (\boldsymbol{M}_j\boldsymbol{W}^{-1})(\boldsymbol{W}\boldsymbol{P}_i), \quad i = 1,\ldots,p, \quad j = 1,\ldots,v. \tag{6}$$

As a matter of fact, if $\boldsymbol{M}_j$ and $\boldsymbol{P}_i$ are a solution, so are $\boldsymbol{M}_j\boldsymbol{W}^{-1}$ and $\boldsymbol{W}\boldsymbol{P}_i$ where $\boldsymbol{W}$ is a collineation of the 3D space, i.e. a 4×4 invertible matrix.

Therefore, a basis for any 3D collineation can be arbitrarily chosen in the 3D space. For the projective space $\mathcal{P}^3$, five algebraically independent points (no four of them being coplanar) form a basis.

2.2.2. Direct nonlinear reconstruction

From Eqs. (5), the problem can be formulated as a nonlinear optimization problem. In the general case we have to estimate parameters (here the matrices $\boldsymbol{M}_j$ and the 3D coordinates of the points $\boldsymbol{P}_i$) having noisy measurements (here the image coordinates). We assume that the measurements are obtained with a mean value equal to the observed one, and with a covariance matrix $\boldsymbol{C}$.

Let us call $\boldsymbol{Q}$ the vector of all parameters, and q_k an element in $\boldsymbol{Q}$, $\boldsymbol{U}$ the vector of all the measurements U_{ij} and V_{ij}, and let u_l be one of its elements.

If the relation between the measures u_l and parameters q_k is linear, i.e. $\boldsymbol{U} = \boldsymbol{A}\boldsymbol{Q}$, then the maximum likelyhood estimation of the parameters is the vector $\boldsymbol{Q}$ which minimizes the Mahalanobis distance, i.e. the least square criterion

$$\chi^2 = (\boldsymbol{U} - \boldsymbol{A}\boldsymbol{Q})^{\mathrm{T}}\boldsymbol{C}^{-1}(\boldsymbol{U} - \boldsymbol{A}\boldsymbol{Q}). \tag{7}$$

In the nonlinear case, linearization may be obtained by taking the first-order of Taylor expansion of the nonlinear function linking $\boldsymbol{Q}$ with each u_l.

Noise is usually due to many independent causes. The perspective model is wrong because of the existence of optical distortions, and more important are errors in image point positions. The latter are highly uncorrelated, while the former is not. So in general,

the major errors can be considered as an uncorrelated noise. Since the covariance matrix of the noise due to other sources is less important and very hard to estimate, it was not taken into account.

As a consequence our covariance matrix can be taken to be diagonal, with values equal to the variances σ_{ij}. Therefore, in our case, Eq. (7) leads to the minimization of this simple sum:

$$\chi^2 = \sum_{ij} \left(\frac{U_{ij} - \dfrac{m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}}{\sigma_{ij}} \right)^2 + \sum_{ij} \left(\frac{V_{ij} - \dfrac{m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}}{\sigma_{ij}} \right)^2 .$$

Section 4 provides details of the method used to solve this system and also presents the reconstruction results.

When the Euclidean structure (Euclidean coordinates) of the five points used as a basis is known, we obtain a Euclidean reconstruction of the scene, (Section 4). On the other hand, if no Euclidean information is provided, the only kind of 3D reconstruction that can be obtained is projective, i.e. the solution has no metric information and is defined up to a collineation.

For many applications Euclidean information is not necessary, we will show in the next section how to derive some symbolic relations using only projective information and geometrical constraints. We will also present an alternative technique for obtaining the Euclidean information, using knowledge about the scene instead of known 3D point positions.

3. Computing symbolic relations

This section discusses the symbolic relationships that can be computed from various kinds of geometric information. We assume that the projective reconstruction of the scene has been computed from at least two views.

3.1. Relations in the quasi-affine space

This section is inspired by a similar work done by [24]. It is also related to the cheirality invariants described by Hartley [8]. Both approaches allow the space to be oriented.

Let A and B be two distinct points in the projective space. The line (A, B) passing through A and B is defined to be the set of points with coordinates equal to $\lambda A + \mu B$, λ and μ being defined up to a scaling factor as usual. However as was already explained in

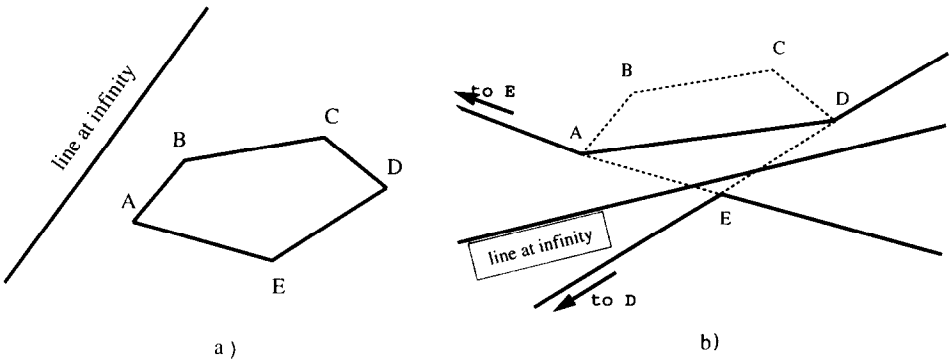

Fig. 2. Influence of the ideal line position on the convex hull.

Fig. 1, this does not allow the segment $[A, B]$ to be uniquely defined as two segments are possible.

The segment $[A, B]$ can only be defined using affine information: if the connected part of the line limited by A and B does not contain an ideal point. From this definition it is clear that the segment remains identical when the ideal point moves continuously along the line without passing A or B; for instance the bold part of Fig. 1 is still the bold segment if the ideal point is located in C or D. An approximate location for the ideal point would be sufficient to express this kind of topological information, and the related relationships "*the point is between A and B*".

The convex hull problem is similar. Fig. 2 illustrates the behavior of the convex hull for two different affine interpretations of the plane, i.e. for two different lines chosen as being the line at infinity. In the first case, as the line at infinity (the ideal line) does not pass through the intuitive convex hull (Fig. 2(a)), the convex hull remains similar to the intuitive one. But in Fig. 2(b) this line cuts the intuitive convex hull, leading to the fact that B and C are inside the hull for this case. No side of the convex hull cuts the infinity line. Notice that in this case, we only need to know whether the ideal line passes through the true convex hull or not; this naturally extends to the ideal plane in the 3D space (see [24]).

In fact, we have additional information in the image as we already know its line at infinity. This line back-projects to a plane passing through the center of projection and parallel to the image plane. This needed plane allows us to compute the convex hull. If M_3 is the third row of the projection matrix M, the equation of the plane is

$$M_3 \cdot (x, y, z, t)^{\mathrm{T}} = 0.$$

A practical implementation for the computation of the convex hull is to choose an arbitrarily coordinate frame so that $t = 0$ is the equation of the plane which simulates the plane at infinity. In this coordinate frame, the unit on the axis are considered as an Euclidean basis which therefore allows to use all the known algorithms for convex hull (see [21] for having a collection of efficient algorithms). From there the definition of the relation *inside* is straightforward: the convex hull introduces a partition between inside and outside.

3.2. Relations in the affine space

The ratio of aligned segments is the main invariant of affine geometry. Hence, saying that a point is exactly in the middle of two others is an affine definition. More generally, if not only an *insideness* relation is needed, but also the location of where inside, then quasi-affine space is no longer sufficient and at least affine space is required.

As we have already mentioned, the affine space can be deduced from the projective space when the position of the ideal plane Π_∞ is known. This allows ratios of parallel segments to be computed, and therefore relations like

- *closest along a given direction*: compute the ratio of lengths in the given direction
- *center of*: here center might be the center of gravity, the exocenter of a triangle,. . ..

The ideal plane can be defined in the projective space by at least three noncollinear points at infinity. The most common way to get such points, is to consider lines that are known to be parallel and therefore intersect at infinity in the 3D space. In the images, these points correspond to vanishing points. If three pencils of parallel lines can be found in the images, they define three vanishing points which are enough to get the ideal plane.

An alternative to this is to consider the basic affine invariant: the ratio of three collinear points A, B, C. It should be noticed that this ratio is related to the cross ratio of the four points A, C, B, D where D is the point a infinity on the line $[A, B]$, as

$$[A, B; C, D] = \frac{AC}{AD} \times \frac{BD}{BC} = \frac{AC}{BC} \times \frac{BD}{AD} = \frac{AC}{BC} \times \frac{\infty}{\infty} = \frac{AC}{BC}.$$

If we know the projections of these points in the images, we are therefore able to derive from the value of this ratio the image locations of D and therefore to reconstruct its position in the projective space.

In structured worlds some numerical information is often available about the position of the observed points, for instance A may be the midpoint of B and C. As the midpoint is the conjugate (i.e. with cross ratio equal to -1) of the ideal point, and as the cross ratio of point on a line is conserved by projection, the projection of the ideal point is easily computed when projections of A, B and C are known.

Once the ideal plane is located in projective space, computing affine information is straightforward. Stating that Π_∞ should have equation $t = 0$ yields three independent linear equations which constrain the last line of the matrix W (6), one equation for each ideal point (the projective reconstruction is defined up to a collineation W).

3.3. Relations in other geometric spaces

Some relationships use geometric information that fits exactly into the standard hierarchy of geometries: projective, affine and Euclidean. *Above* is an example of such a relationships and uses several features. One is the natural vertical direction not present in the affine space. A second is that the object A should project itself vertically onto object B, or that the projection of A overlaps B.

To deal with a direction we need to know the corresponding ideal point. This can be found using one of the previously described techniques. It can also be done using

camera orientation information. Notice that if the camera is known to have no roll, *a priori* information is available: the central vertical line of the image back projects to a vertical plane: from two images a vertical line can then be extracted. However this allows only reasoning in an area close to this line, as vertical lines might project to oblique lines in the image if the principal axis of the camera is not horizontal.

3.4. Similarity relationships

Measurements of absolute size are often not needed for spatial reasoning. For instance, if we want to know whether an object will pass through a hole, it is sufficient to know the *relative* size of the hole and the object. Absolute measurements are unnecessary. In this case objects can be defined up to a uniform scaling factor and no Euclidean positions are needed.

This leads us to consider the group of similarities, which preserves angles and ratios of lengths. To get a reconstruction up to a similarity, new information has to be added to the affine reconstruction. Mathematically, this can be done by finding an affine transformation $\mathcal{A}$ such that the *a priori* information is satisfied in the transformed space. Such a numerical frame is general but not easy to implement.

A more elegant way to recover a suitable affine transformation is to consider the absolute conic Ω.

The absolute conic is located in Π_∞ [16,25] and defined by the equations:

$$\begin{cases} x^2 + y^2 + z^2 = 0, \\ t = 0, \end{cases} \tag{8}$$

where the previous equations are expressed in an "extended" Euclidean reference frame, i.e., an orthogonal reference frame with similar length unit on each axis. All points on this conic have complex coordinates. From direct computation it is straightforward to check that Ω is invariant by uniform scaling, translation and rotation.

The easiest way to determine this conic is to reconstruct known circles in the 3D space. Each circle intersects Π_∞ in two points belonging to Ω [25]. From three such circles the reconstruction of Ω is possible. Let its equation be:

$$a_1x^2 + a_2y^2 + a_3z^2 + 2a_4xy + 2a_5xz + 2a_6yz = 0.$$

A change in coordinates is necessary to bring it into the form of Eq. (8). This can be done by considering that the equation of the conic is associated with the quadratic form Q defined by

$$0 = (x, y, z) \begin{pmatrix} a_1 & a_4 & a_5 \\ a_4 & a_2 & a_6 \\ a_5 & a_6 & a_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = X^{\mathrm{T}}QX.$$

As the matrix Q is symmetric, there exists an orthogonal matrix P such that

$$Q = P^{\mathrm{T}} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} P.$$

So setting $X' = PX$, we have:

$$X^{\mathrm{T}} Q X = (X')^{\mathrm{T}} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} X'.$$

Finally, with a further rescaling along each axis, we get Eq. (8).

Another way to proceed is to use the basic extended Euclidean invariant: the value α of the angle formed by two coplanar lines L and L'. This invariant can also be used to compute Ω, as knowledge of it introduces a constraint on its location. Let A and A' be the intersections of the two lines with Π_∞. Let I and J be the intersections of the plane defined by these two lines with Ω. Laguerre's formula states that:

$$\alpha = \frac{1}{2\mathrm{i}} \log(\{A, A'; I, J\}).$$

We can write $I = A + tA'$ and $J = A + t'A'$. With this notation

$$\{A, A'; I, J\} = t/t' = \mathrm{e}^{2\mathrm{i}\alpha}.$$

If we express that both points lie on Ω we get the equations

$$\begin{aligned} &t^2 A'^{\mathrm{T}} Q A' + 2t A^{\mathrm{T}} Q A' + A^{\mathrm{T}} Q A = 0, \\ &\mathrm{e}^{4\mathrm{i}\alpha} t^2 A'^{\mathrm{T}} Q A' + 2\mathrm{e}^{2\mathrm{i}\alpha} t A^{\mathrm{T}} Q A' + A^{\mathrm{T}} Q A = 0, \end{aligned} \tag{9}$$

from which a polynomial constraint on Q is easily derived: elimination of t^2 in the previous equations leads to

$$2t A^{\mathrm{T}} Q A' \beta(\beta - 1) + A^{\mathrm{T}} Q A(\beta^2 - 1) = 0 \quad \text{with } \beta = \mathrm{e}^{2\mathrm{i}\alpha},$$

t can be extracted from the above

$$t = -\frac{(\beta + 1)}{2\beta} \frac{A^{\mathrm{T}} Q A}{A^{\mathrm{T}} Q A'}.$$

Substituting t in (9) provides a quadratic polynomial constraint on Q

$$\left(\frac{1 + \beta}{2}\right)(A^{\mathrm{T}} Q A)(A'^{\mathrm{T}} Q A') - (A^{\mathrm{T}} Q A')^2 = 0.$$

Therefore, the absolute conic Ω can be computed given five angles, but we did not succeed to get a solution based on the above equation. Hence, this theoretical discussion seems irrelevant since for our experiments we have used different Euclidean constraints. However, the above discussion shows clearly the relationships between projective, affine and Euclidean reconstruction. Furtheremore, we have derived exactely what is needed for each kind of reconstruction.

4. Implementation and results

4.1. Implementation

This section explains our reconstruction procedure in detail, gives examples of its use on both real and simulated data, and discusses the accuracy of the reconstruction results.

4.1.1. 3D reconstruction

The reconstruction problem is specified by the set of equations (5), as mentioned in Section 2.2.2, the resolution of the problem turns into a nonlinear optimization problem. Since the projective coordinates of the spatial points are defined up to scaling factor, the following constraint is added

$$x_i^2 + y_i^2 + z_i^2 + t_i^2 = 1. \tag{10}$$

A similar scaling constraint has to be imposed on the projection matrices. Usually the following constraint is used:

$$\sum_{i=1}^{3} m_{3i}^2 = 1.$$

However, since this condition was not crucial for the convergence, the simpler constraint $m_{34} = 1$ was used instead (we know that m_{34} is not null in our formulation).

Finally, Eqs. (5) were homogenized leading to the minimization of the square of

$$U_{ij} \times (m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + t_i) - (m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i) \tag{11}$$

and

$$V_{ij} \times (m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + t_i) - (m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i) \tag{12}$$

together with

$$x_i^2 + y_i^2 + z_i^2 + t_i^2 - 1. \tag{13}$$

To fit with the Levenberg–Marquardt formulation [22], the problem is written as a minimization of

$$\sum_{k=1}^{2\times p\times v+p} \left(\frac{f_k(x_{ij}, y_{ij}; x_i, y_i, z_i, t_i, m_{11}^{(j)}, \ldots, m_{34}^{(j)})}{\sigma_k} \right)^2 \tag{14}$$

over the parameters

$$(x_i, y_i, z_i, t_i, m_{11}^{(j)}, \ldots, m_{33}^{(j)}) \quad \text{for } i = 1, \ldots, p, \quad j = 1, \ldots, v;$$

where σ_k represents the variance of the kth measure, and $f_k(\cdot)$ the functions (11), (12) or (13).

For such nonlinear optimization, initialization is a crucial step. Even though the method converges for most of our examples with a very "ad hoc" initialization, this can

not be guaranteed. Therefore, we initialize the 3D structure from a parallel projection model [30,32]. This initialization is not too far from a real reconstruction even with large perspective distortion (see [3] for details). Reconstruction is therefore done in two steps. First, a linear reconstruction is obtained assuming a weak perspective projection [32], then, this reconstruction is used as an initial guess to our nonlinear algorithm. Note that unlike the orthographic projection model, the weak perspective one has no restriction on the camera motion (the camera can move forward the scene).

4.1.2. Estimation of the collineation W

Once we have a projective reconstruction defined up to a collineation W, we need to recover the Euclidean structure without knowing any 3D point, i.e., we need to find a transformation W which satisfies the added Euclidean constraints as well as possible to bring the solution to Euclidean space.

The collineation W can also be constrained using knowledge from the camera. For example by assuming that the camera internal parameters do not change and that the plane at infinity can be estimated. However, the performance of such algorithm is not satisfactory yet and a large number of of images is needed to make the computation of W stable [9]. In fact, when the camera internal parameters remain constant, given at least 3 images of the same scene provide enough constraints to get Euclidean reconstruction [13]. Unfortunately, these constraints translate in very high nonlinear equations.

The problem here is much easier since it only concerns the estimation of the 15 unknowns of W. Again, since W is defined up to a scaling factor, we add a constraint

$$\sum_{i,j} (W_{ij})^2 = 1.$$

Similarly, the problem can be written as a minimization over:

$$\sum_{k=1}^{n} \left(\frac{f_k(W_{11}, \ldots, W_{44})}{\sigma_k} \right)^2, \tag{15}$$

where $f_k(\cdot)$ are functions describing the particular geometric constraints as we will see with a real application in the next subsection.

4.2. Results

First, the algorithm using five points in the scene as a relative reference frame is evaluated with real data and the use of geometrical constraints is illustrated with a real example. Then, a quantitative comparison between the Euclidean reconstruction obtained with five known points and the one obtained using geometrical constraints is made using simulated data. Finally, an example of the symbolic relationships *inside* and *above* is given.

4.2.1. Reconstruction results using five points

Three different scenes have been used here:

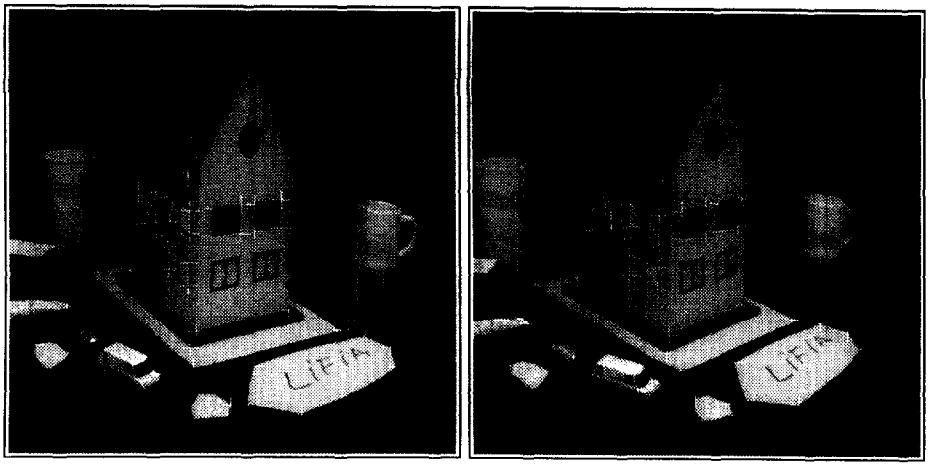

Fig. 3. The house scene: the three images used for the reconstruction together with the extracted corners. The relative reference frame used for the reconstruction is defined by the five points marked with white disks in image 3.

- The wooden house scene (Fig. 3). The size of this indoor scene is 40 cm × 40 cm × 25 cm, it is located at about 1 meter from the camera. Three images of this scene, covering about a $\pi/6$ rotation of the camera, have been used. A total of 49 points have been extracted using our corner detector [17], and tracked over the images. Five points on the house have been measured (with an ordinary ruler) and chosen as a basis. These are denoted by disks in Fig. 3. The difference between the reconstructed point coordinates and the measured point coordinates was less than 1.5 mm. Fig. 4 shows the reconstructed scene from several different points of view.

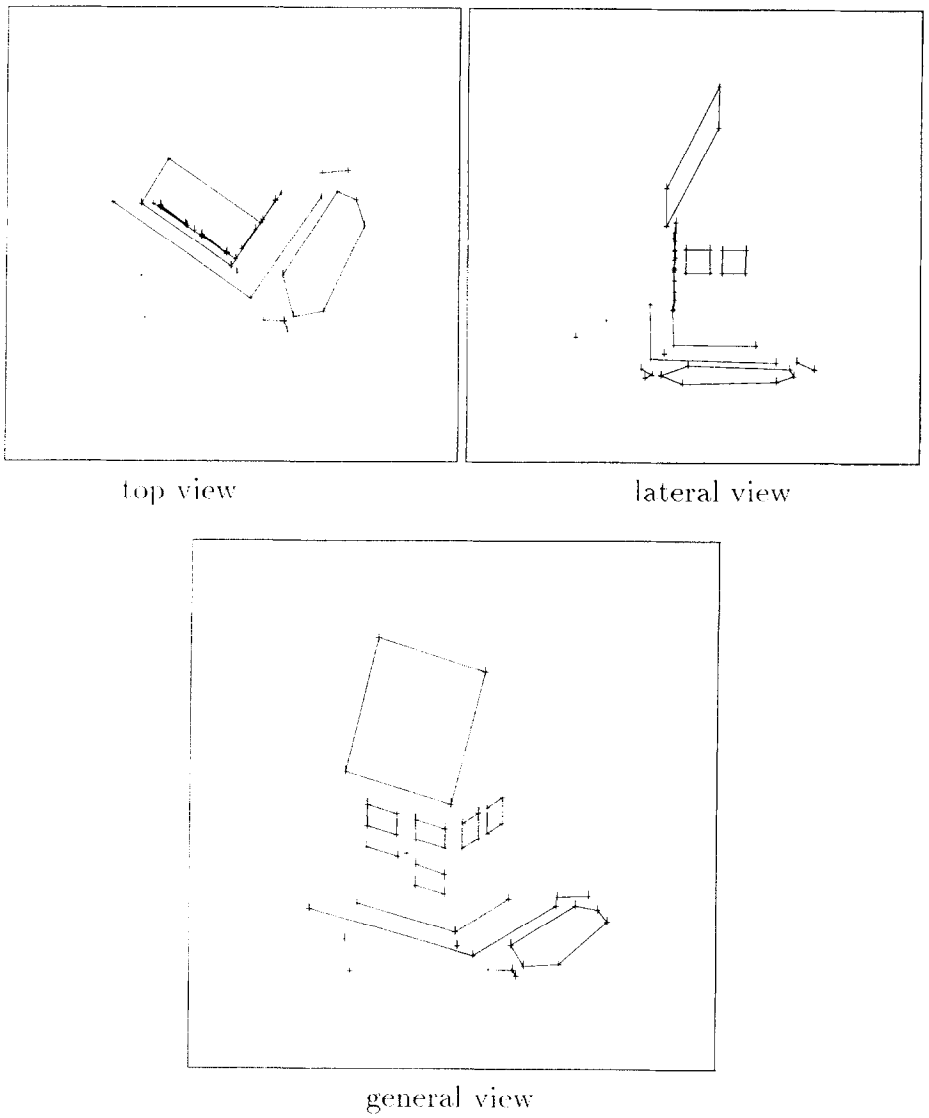

Fig. 4. Euclidean reconstruction of an indoor scene using five known points. The reconstructed points are joined with segments to make the result more expressive.

- The outdoor scene (Fig. 5). Three images of an outdoor scene have been taken with a hand-held camera. A total of 162 points were extracted and tracked automatically over the three images. Five points were used as a basis, they are denoted by thick crosses in Fig. 5. Since we have no exact Euclidean coordinates in this case, we assigned approximate coordinates to the five points used as a basis. Although the reconstruction obtained is projective, however, it is quasi-Euclidean

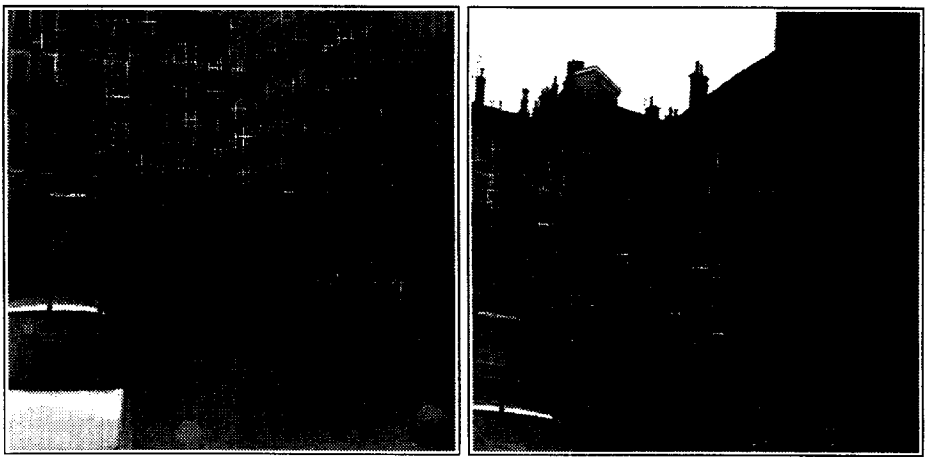

Fig. 5. Three images of an outdoor scene taken with a hand-held camera. Unlike the wooden house case, the motion of camera is not regular. The five points defining the basis are marked with thick crosses in image 3.

since the basis coordinates are an approximation to the Euclidean ones. Therefore, the reconstruction has some Euclidean aspects (see Fig. 6).

- The calibration pattern (Fig. 7). This last experiment aims at measuring the accuracy of the reconstructed points. The goal is to reconstruct a set of points which are known to be coplanar, then to check their 3D coplanarity.

  This scene consists of three planes with 160 targets in a volume of diameter 55 cm. The points to be reconstructed are the target centers. We extract such points by computing the best affine transformation between a theoretical target and the

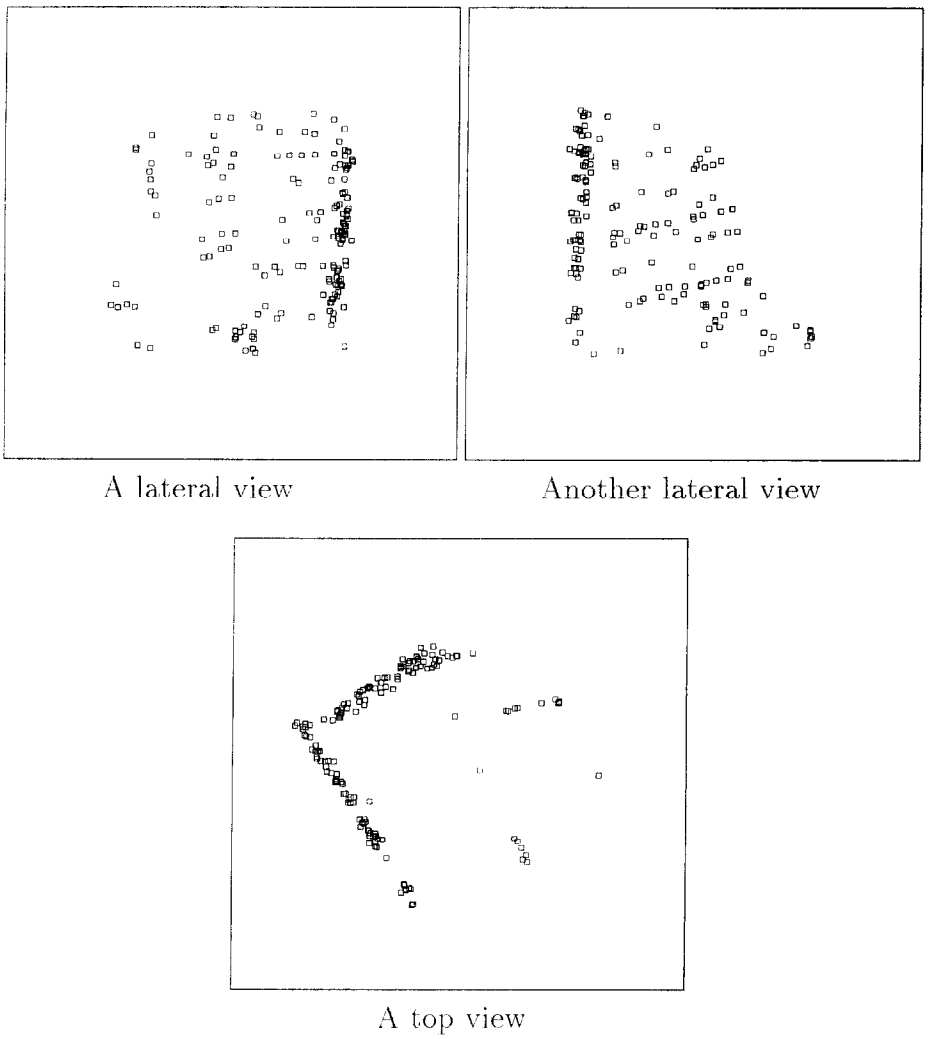

Fig. 6. The result of the reconstruction algorithm for the outdoor scene, the top view is more expressive since the two walls are clearly identified.

real one. The accuracy of the extracted target centers is one tenth of a pixel. Four images were u sed for the reconstruction process. The result is shown in Fig. 8, note that points expected to be coplanar form nearly a perfect plane.

The reconstructed points belonging to the horizontal plane of the calibration pattern were fitted into a plane (the choice of the horizontal plane is arbitrary, we can select any other plane of the calibration pattern). The mean distance of the points to this plane was 0.12 mm and the maximum distance is 0.49 mm. This quantitative result shows that we can reach a very high accuracy in the reconstruction with real images. This is due to the optimal nature of our reconstruction

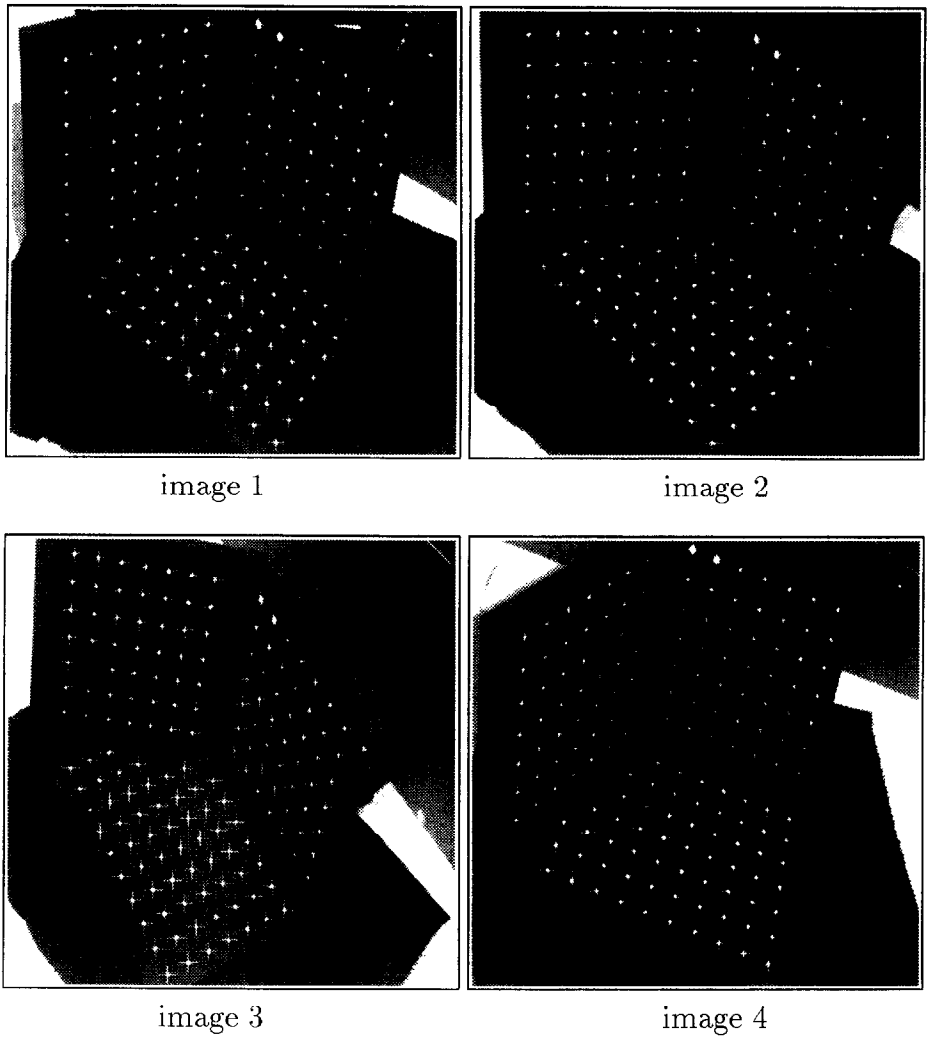

Fig. 7. Calibration pattern image sequence: 4 images were used for the reconstruction process. The interest points are the target centers.

algorithm and to the localization of the image points to subpixel accuracy (one tenth of a pixel in this case).

4.2.2. Using geometric constraints for a real case

The Euclidean reconstruction shown in Fig. 4 was obtained using five known points as a relative reference frame. Now we consider the same scene, but instead of using five known points we will use geometric constraints to obtain the Euclidean reconstruction. The projective reconstruction is first computed using five points as a projective reference frame, then geometric constraints are used to estimate the collineation W which brings

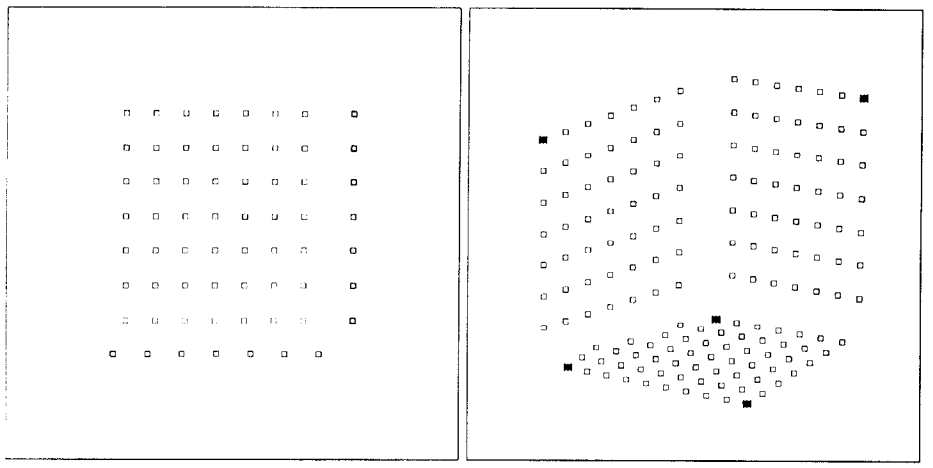

Fig. 8. The reconstructed points of the calibration pattern. Five points are marked with a dark squares on the "general view": they form the relative reference frame.

this projective reconstruction to a Euclidean one.

The constraints of the real problem do not exactly fit the steps described in Section 3. However, understanding these steps helps to avoid underconstrained systems. For instance, consider an image containing a horizontal plane, in which points are indicated as belonging to this plane. In addition, suppose that vertical lines are provided. Thus, we define the plane $z = 0$ and the point at infinity on the z-axis. Obviously, the plane Π_∞ is not defined and the solution is still only defined up to a projective transformation.

Suppose that, for a scene like the one shown in Fig. 3 the three main directions are given. This provides us with three points at infinity and hence defines Π_∞. The points at infinity correspond to vanishing points in the images [23]. Given this, the reconstruction is defined up to an affine transformation. Furthermore, if these directions are orthogonal the rotational part will be defined. However, the absolute conic is not defined yet: at least two more constraints are needed for fixing the conic's five degrees of freedom. If the scaling on the three axes are provided, then the uniform scaling measure is set, as are the two remaining degrees of freedom for the absolute conic. The reconstruction is then defined up to a similarity.

Our example

As an example, consider the previous wooden house scene.

The constraints we applied were a mixture of affine and Euclidean ones.

To get a unique Euclidean solution we fixed a reference frame in the scene in which the constraints were expressed. In our opinion, this is not a great restriction as in almost all the scenes we can find such a reference frame. In most indoor and outdoor scenes there is a floor that can be used as the horizontal plane *XOY*. Also two vertical planes that are perpendicular to each other can be found (building walls, room walls, etc.), we can use these as the *XOZ* and *YOZ* planes respectively.

In the following we must keep in mind that we have a reference frame, but no coordinates are given.

We use the following notation. $A' = (x_{A'}, y_{A'}, z_{A'}, t_{A'})^T$ is the vector of a point before adding the geometrical constraints, and $A = WA' = (x_A, y_A, z_A, 1)^T$ the vector of the point with corrected coordinates (Euclidean coordinates). W is the 4×4 matrix representing the projective transformation to be computed. The following constraints were used:

- the reference planes were fixed,
- points alignment with the axes,
- distances between points.

Result

We used the same 49 points (Fig. 3). The projective reconstruction was computed by solving the nonlinear system (5) using five points as a projective reference frame. Finally, the collineation W was estimated using the following constraints:

- The floor was used as the horizontal plane for our Euclidean reference frame. If a point $A' = (x_{A'}, y_{A'}, z_{A'}, t_{A'})^T$ is known to belong to this plane then $z_A = 0$. This gives a linear equation constraining W.
- The two walls of the house were used as the planes (XOZ) and (YOZ). Points belonging to these planes give rise to the same kind of constraints as for the horizontal plan.
- Up to now, the three planes of our reference frame have been fixed. The axes are obtained from the intersection of theses planes and if a pair of points forms a line parallel to one of the three axes we obtain an alignment constraint.
- Metric information is required to get a full solution. We used distances between pairs of points aligned with an axis. This yields a second-degree equation.

Once W has been estimated, the projective reconstruction is transformed into a Euclidean one. The result is qualitatively similar to the one obtained with five known points (Fig. 4). Quantitative evaluations will be given in the next subsection.

4.2.3. Accuracy in reconstruction

Here we use simulated data to discuss the accuracy of the reconstruction and compare the result based on five known points with the one based on constraints.

We simulated a scene similar to our wooden house scene. Four images covering about a $\pi/2$ rotation of the camera were simulated with 60 points, and the same type of constraints were used.

Table 1 shows the 3D coordinates of 10 points of the simulated scene, the errors when using five known points and the errors when using constraints. Pixel uncertainty was generated as a uniform noise with a maximum deviation of 1 pixel.

From these results, it can be seen that the errors in the 3D coordinates are reasonable but not good enough to conclude that the method is robust when noise is present. In order to study the stability of the method, different levels of noise were added to our data (2D coordinates), then 3D coordinates are computed with the two methods. Fig. 9 displays the mean e rrors on 3D positions when perturbing the images. Of course both are perfect without noise, up to tiny numerical round-off errors. According to Fig. 9,

Table 1
Example of reconstructed 3D coordinates, and the errors.

3D coordinates			Errors (using five points)			Errors (using constraints)		
X	Y	Z	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
12.000	0.000	0.500	0.047	0.073	0.178	0.041	0.059	0.004
0.000	18.000	14.500	0.178	0.013	0.018	0.000	0.025	0.042
6.000	0.000	22.500	0.088	0.061	0.149	0.054	0.018	0.069
12.000	0.000	14.500	0.249	0.065	0.020	0.000	0.025	0.042
13.500	−1.500	0.500	0.068	0.085	0.004	0.011	0.026	0.023
−1.500	−1.500	0.500	0.010	0.107	0.068	0.011	0.026	0.023
−1.500	19.500	0.500	0.048	0.072	0.042	0.038	0.001	0.005
13.500	−1.500	0.000	0.062	0.083	0.036	0.002	0.032	0.000
0.000	6.000	0.500	0.062	0.023	0.045	0.000	0.007	0.054
0.000	12.000	0.500	0.001	0.016	0.020	0.026	0.014	0.007

redundant Euclidean constraints provide a better accuracy in the reconstruction. However, it has to be noted that the two methods for obtaining the Euclidean reconstruction do not use the same Euclidean information. The method using five known points is sensitive to the location of these five points in the scene while the method using constraints is sensitive to the number and kind of the used constraints.

With larger noise amplitudes, results degrade quickly [19]. If accuracy in 3D positions is needed, subpixel measurements are essential, even with redundant geometrical constraints.

4.2.4. Example of symbolic relationships

- *Inside.* With an approximation of the ideal plane by assuming an affine transformation for the scene-image projection, the affine (quasi-affine) reconstruction of the scene was performed, then, the convex hull was computed using a subset of these reconstructed points. Fig. 10 shows the projection of the convex hull on the image, the points of the scene split into two subsets: points *inside* the *convex hull* and points outside.
- *Above.* This relation needs the knowledge of the vertical direction which in our case is computed by finding in the different images (in at least two images) the vanishing points corresponding this direction (Fig. 11). Therefore, the ideal point in the 3D projective space is computed, and the relation *above* between a point A and an object B consists simply of testing whether the 3D line AC intersects the object B, where C is our ideal point.

5. Conclusion

This paper shows how a scene can be reconstructed from multiple uncalibrated images using a parameter estimation approach. The reconstruction is relative, i.e. point coordinates are computed relatively to five points in the space. Since an absolute reference

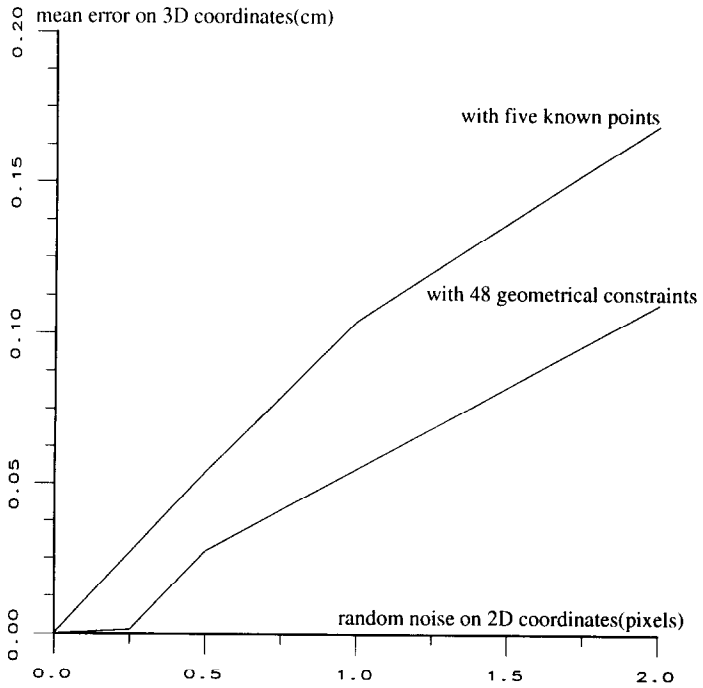

Fig. 9. Since the noise is important the method using constraints is more suitable.

frame is not needed for the method, the exact positions of the five points do not need to be known.

This an approach allows us to work with a camera with automatic focus and aperture, without knowing the position from where each view is taken and without knowing the internal camera parameters. However, if the only information is the point matches between the images, the reconstruction can only be computed up to a projective transformation of the 3D space. Therefore, if more than projective information is needed, geometric constraints have to be added. The parameter estimation framework presented in this paper allows a wide variety of geometric constraints to be incorporated into the reconstruction algorithm. In particular, Euclidean reconstructions can be obtained from projective ones if enough constraints are available.

The resolution of the nonlinear reconstruction algorithm is done in two steps, first, an approximation to the solution is obtained assuming an affine model for the camera, then, this solution is used to initialize the nonlinear algorithm.

It was shown that our projective reconstructions contain more than just projective information. With complementary information such as an estimate of the relative location of the plane at infinity, some spatial reasoning is possible. Qualitative questions like "*is this point inside a set of points?*" can be then answered.

If more information about the environment is available, further symbolic relations containing richer information can be derived. Since in many applications the environment is partially known, the framework presented in this paper is likely to prove very

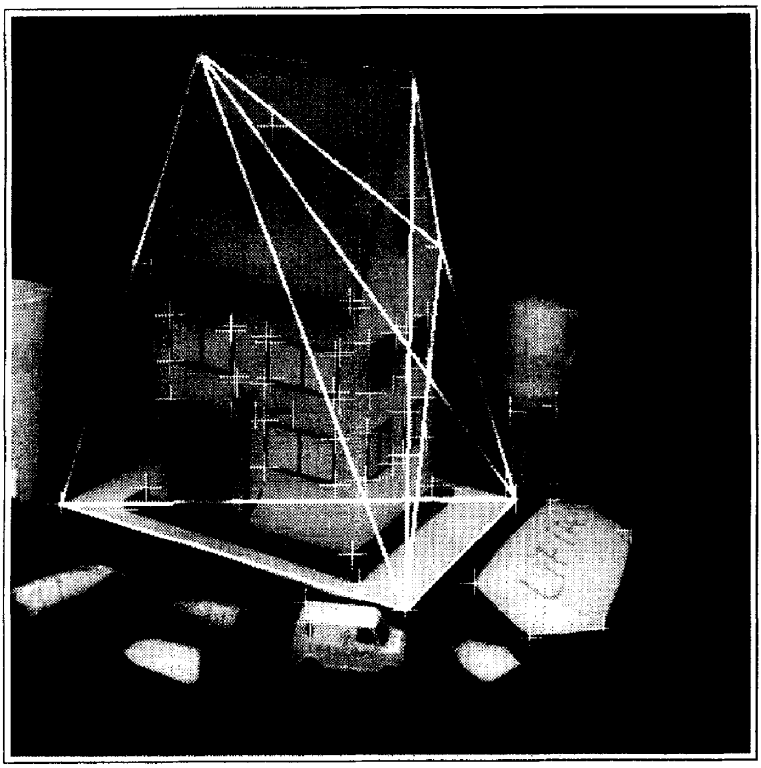

Fig. 10. The computed convex hull projected on the image, the relation *inside* is defined relative to this convex hull.

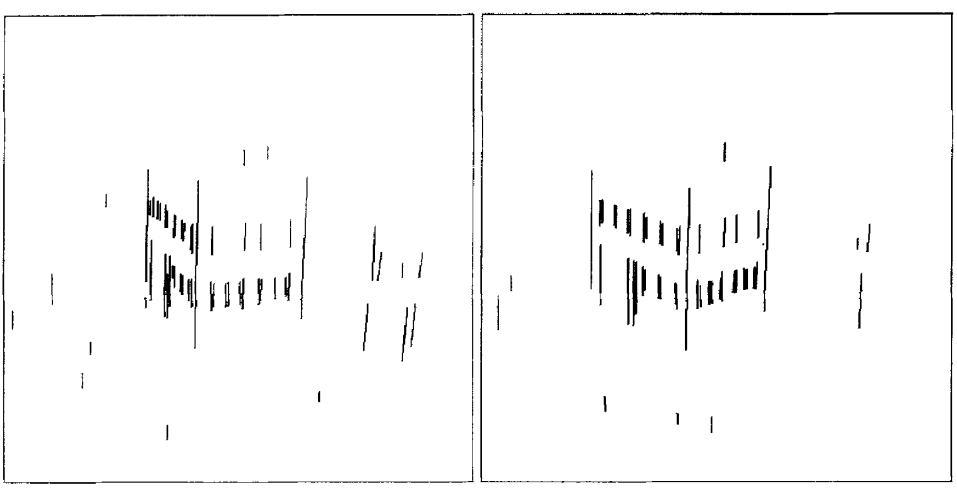

Fig. 11. The vertical direction in two images computed by finding the vanishing points in that direction.

promising in the near future. Therefore, our future work will focus on further types of constraints we can extract from realistic environments, and how these constraints can be combined with image measurement to get useful information.

References

[1] A. Azarbayejani, B. Horowitz and A. Pentland, Recursive estimation of structure and motion using relative orientation constraints, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, New York (1993) 294–299.

[2] H.A. Beyer, Accurate calibration of CCD cameras, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, Urbana-Champaign, IL (1992) 96–101.

[3] B. Boufama, D. Weinshall and M. Werman, Shape from motion algorithms: a comparative analysis of scaled orthography and perspective, in: J. Eklundh, ed., *Proceedings 3rd European Conference on Computer Vision*, Stockholm, Sweden (Springer-Verlag, Berlin, 1994) 199–204.

[4] B.A. Draper and E.M. Riseman, Learning 3D object strategies, in: *Proceedings 3rd International Conference on Computer Vision*, Osaka, Japan (1990) 320–324.

[5] N. Efimov, *Advanced Geometry* (Mir, Moscow, 1978).

[6] O.D. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig?, in: G. Sandini, ed., *Proceedings 2nd European Conference on Computer Vision*, Santa Margherita Ligure, Italy (Springer-Verlag, Berlin, 1992).

[7] J.A. Feldman and Y. Takimovsky, Decision theory and artificial intelligence I: a semantic based region analyser, *Artif. Intell.* **5** (4) (1974) 349–371.

[8] R.I. Hartley, Cheirality invariants, in: *Proceedings DARPA Image Understanding Workshop* (1993) 745–753.

[9] R.I. Hartley, An algorithm for self calibration from several views, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, Seattle, WA (1994) 908–912.

[10] R.I. Hartley, R. Gupta and T. Chang, Stereo from uncalibrated cameras, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, Urbana-Champaign, IL (1992) 761–764.

[11] J.J. Koenderink and A.J. van Doorn, Affine structure from motion, Tech. Rept., Utrecht University Utrecht, Netherlands (1989).

[12] C.H. Lee and T. Huang, Finding point correspondences and determining motion of a rigid object from two weak perspective views, *Comput. Vision Graph. Image Processing* **52** (1990) 309–327.

[13] Q.T. Luong and O.D. Faugeras, Self-calibration of a camera using multiple images, in: *Proceedings 11th International Conference on Pattern Recognition*, The Hague, Netherlands (1992) 9–12.

[14] Q.T. Luong and T. Vieville, Canonic representations for the geometries of multiple projective views, in: *Proceedings 3rd European Conference on Computer Vision*, Stockholm, Sweden (1994) 589–599.

[15] S.J. Maybank, The projective geometry of ambigious surfaces, Tech. Rept. 1623, Long Range Laboratory, GEC Wembley, Middlessex, England (1990).

[16] S.J. Maybank and O.D. Faugeras, A theory of self calibration of a moving camera, *Int. J. Comput. Vision* **8** (2) (1992) 123–151.

[17] R. Mohr, B. Boufama and P. Brand, Accurate projective reconstruction, in: *Proceedings DARPA-ESPRIT Workshop on Applications of Invariants in Computer Vision*, Azores, Portugal (1993) 203–227.

[18] R. Mohr, L. Morin and E. Grosso, Relative positioning with uncalibrated cameras, in: J.L. Mundy and A. Zisserman, eds., *Geometric Invariance in Computer Vision* (MIT Press, Cambridge, MA, 1992) 440–460.

[19] R. Mohr, L. Quan, F. Veillon and B. Boufama, Relative 3D reconstruction using multiple uncalibrated images, Tech. Rept. RT 84-I-IMAG LIFIA 12, LIFIA-IRIMAG (1992).

[20] R. Mohr, F. Veillon and L. Quan, Relative 3D reconstruction using multiple uncalibrated images, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, New York (1993) 543–548.

[21] F. Preparata and M.I. Shamos, *Computational Geometry, an Introduction* (Springer-Verlag, Berlin, 1985).

[22] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes in C* (Cambridge University Press, New York, 1988).

[23] L. Quan and R. Mohr, Determining perspective structures using hierarchial Hough transform, *Pattern Recognition Lett.* **9** (4) (1989) 279–286.

[24] L. Robert and O.D. Faugeras, Relative 3D positionning and 3D convex hull computation from a weakly calibrated stereo pair, in: *Proceedings 4th International Conference on Computer Vision*, Berlin, Germany (1993) 540–544.

[25] J.G. Semple and G.T. Kneebone, *Algebraic Projective Geometry* (Oxford Science Publication, 1952).

[26] S. Soatto, P. Perona, R. Frezza and G. Picci, Recursive motion and structure estimation with complete error characterization, in: *Proceedings Conference on Computer Vision and Pattern Recognition*, New York (1993) 428–433.

[27] G. Sparr, Projective invariants for affine shapes of point configurations, in: *Proceedings DARPA–ESPRIT Workshop on Applications of Invariants in Computer Vision*, Reykjavik, Iceland (1991) 151–170.

[28] R. Szeliski and S.B. Kang, Recovering 3D shape and motion from image streams using nonlinear least squares, *J. Visual Commun. Image Representation* **1** (5) (1994) 10–28.

[29] J.M. Tennenbaum and H.G. Barrow, Experimentation in interpretation-guided segmentation, *Artif. Intell.* **8** (1977) 241–274.

[30] C. Tomasi and T. Kanade, Factoring image sequences into shape and motion, in: *Proceedings IEEE Workshop on Visual Motion*, Princeton, NJ (IEEE Computer Society Press, Los Alamitos, CA, 1991) 21–28.

[31] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, *IEEE J. Robotics Automation* **3** (4) (1987) 323–344.

[32] D. Weinshall, Model-based invariants for 3-D vision, *Int. J. Comput. Vision* **10** (1) (1993) 27–42.