Understanding the Shape Properties of Trihedral Polyhedra

Charlie Rothwell, Julien Stern

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Understanding the Shape Properties of Trihedral Polyhedra

Charlie ROTHWELL and Julien STERN

N° 2661

Octobre 1995

PROGRAMME 4

ISSN 0249-6399

Understanding the Shape Properties of Trihedral Polyhedra

Charlie ROTHWELL* and Julien STERN**

Programme 4 — Robotique, image et vision
Projet Robotvis

Rapport de recherche n° 2661 — Octobre 1995 — 30 pages

Abstract: This paper presents a framework for the computation of projective invariants of trihedral polyhedra. Invariants for such shapes, which are composed of triples of planes meeting at vertices, were first discussed by Rothwell, *et al.* in [24]. However, they treated only four degree of freedom objects (dof.), and not polyhedra in general. We extend their results to arbitrary dof. figures by showing that more complex shapes can be broken down into sets of connected four dof. polyhedra. Although the more general shapes do not possess projective properties as a whole (when viewed by a single camera), each subpart does yield a projective description. Furthermore, planar projective invariants can be measured which link together the subparts. Consequently, we are able to provide local-global descriptions for general trihedral polyhedra.

The original projective description also involved a difficult mathematical analysis. We demonstrate that a set of *butterfly invariants* can be used in equivalence to the original formulation. We also provide a novel algebraic formulation of the butterfly invariant which is simpler both to implement and to understand than previous approaches. We therefore provide a more straightforward route to the computation of polyhedral invariants.

Finally, we demonstrate the recovery of polyhedral shape descriptions from images by exploiting the local-global nature of the invariants. Local measures can be extracted far more reliably in real images due to the problems of feature segmentation. Additionally, with the aid of a model base, we can extend the local descriptions to global measures and move towards the recognition of entire polyhedra.

Key-words: Polyhedra, geometric invariants, object recognition

(Résumé : tsvp)

*Funded by an HCM grant from the European Community. Email: crothwel@sophia.inria.fr
**Visiting from the Ecole Normale Supérieure de Lyon

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Comprendre la structure des polyèdres trièdraux

Résumé : Dans cet article, nous présentons une méthode de calcul d'invariants projectifs pour des polyèdres trièdraux. Des invariants pour de tels objets ont déjà étés étudiés par Rothwell, *et al.* dans [24]. Cependant, ces auteurs n'ont traité que le cas d'une classe très simple de polyèdres : ceux possédant quatre degrés de liberté. Nous étendons leurs résultats à des polyèdres généraux en montrant que l'on peut diviser ces derniers en un ensemble de polyèdres à quatre degrés de liberté reliés entre eux. Dès lors, même si la figure initiale ne possède aucune propriété projective particulière (lorsqu'elle est vue d'une seule caméra), chaque sous partie peut être décrite projectivement. Qui plus est, on peut calculer des invariants plans qui permettent de positionner les sous figures les unes par rapport aux autres. Ainsi, on parvient à obtenir une description locale et globale de polyèdres trièdraux généraux.

La première étude sur la description projective des polyèdres s'appuyait sur une partie mathématique relativement complexe. Nous montrons ici qu'un ensemble d'invariants "papillons" (butterfly invariants) peut être utilisé à la place de la description précédente. De plus, nous présentons une nouvelle formulation de l'invariant "papillon" qui est plus simple, aussi bien à implémenter qu'à comprendre. Il est donc desormais vraiment plus simple de calculer des invariants pour des polyédres.

Enfin, nous montrons comment obtenir une description de la forme d'un polyèdre extrait d'une images réelles, en exploitant la nature à la fois globale et locale des invariants. Il est clair que les mesures locales peuvent être obtenues avec bien plus de précision que les mesures globales. Ceci est lié aux problèmes de segmentation qui apparaissent sur des images réelles. Cependant, grâce à l'utilisation d'une librairie de modèles, nous parvenons á déduire les mesures globales à partir des descriptions locales, et pouvons ainsi reconnaître n'importe quel polyèdre.

Mots-clé : Polyèdres, invariants géométriques , reconnaissance d'objet

Contents

1 Introduction

In this report we discuss a general scheme which enables the understanding of the shape properties of trihedral polyhedra. Trihedral polyhedra are solid polyhedra made up of a set of planes in arbitrary positions, and as such, there are no special constraints between the planes. The nomenclature trihedral derives from the fact that the vertices of the polyhedra are only ever defined by triples of planes: points in space need at least three planes to assert their locations, but any more than three would provide excess constraint and hence would not be generic (and stably realisable).

1.1 A long history of polyhedral shape recovery

Polyhedral shape recovery has had a long history in computer vision, though with very varied degrees of success. In fact, the programme most often considered to mark the breaking away of computer vision from pattern recognition in the early nineteen-sixties had the goal of segmenting and recognising polyhedra. This was the thesis work of Roberts [22]. Nevertheless, none of the algorithms designed to analyse the simple scenes such as we frequently encounter in computer vision environments have made a mark in real world environments. This is partially due to the fact that simple polyhedra are not as ubiquitous as one might desire. Certainly, many man-made objects are based around polyhedral structures, but they are frequently marred by surface textures, markings, specularities, shadows and other features which impede reliable segmentation. Confound with this that many polyhedra have non-polyhedral subparts, and that scene clutter ceaselessly succeeds in confusing segmentation algorithms, and one is left with little surprise that robust working systems are rare.

However, when put in context there have been a number of promising steps forward. Roberts [22] was able to recover the identity and pose of polyhedral objects from scanned images. This early achievement was followed in the nineteen-seventies by a number of programmes investigating the forms which images of polyhedra can take. Notionally these expressed the geometric constraints of the imaging process when applied to polyhedra as a grammar for parsing images. The results of these investigations allowed the interpretation of line drawings of polyhedra and hence an understanding of which interpretations may or may-not be correct. The most notable of these were the projects of Clowes [7], Huffman [14], Mackworth [18] and Waltz [31] and Guzman [12]. Although it is now widely accepted that recovering the complete line drawings required by these algorithms is not practicable, the basic rules of reasoning still hold an important place in the geometric understanding of scenes today.

Subsequent to the grammatical approaches, Sugihara [28] built an algebraic framework on top of the fundamental geometric facts. Through algebraic reasoning, he was able to provide much stronger constraints than previously on the three-dimensional shapes of observed polyhedra in space. This is because some projections which satisfy the rules defined by the grammars are neither realisable in practice, nor satisfy the algebraic constraints (both the scientific and pictorial arts communities have exploited thoroughly the world of *impossible drawings*). The final result of Sugihara's investigation is that through the use of camera

calibration and shading information estimated from the surface planes of polyhedra, one can produce complete three-dimensional reconstructions for viewed polyhedral scenes.

All of the early approaches suffered from the same basic drawback: they required accurate recovery of the topology of the line drawings of polyhedra. In practice, the generally poor performances of edge detectors around object junctions and the invariable presence of low contrast internal edges of polyhedra means that the basic line descriptions cannot be recovered. This is the case even without the presence of clutter in scenes. For some uncomplicated scenes, as demonstrated very clearly by Roberts, one can make sufficient use of context from the model base to complete the silhouettes and other topological boundaries in images. However, more generally the degree of search required to achieve the task can become insurmountable. Consequently the extraction of sufficient topological structure for the recognition of polyhedra is not only difficult, but nigh impossible in real images.

That being said, there have been a number of more recent successes through the use of more general strategies. The algorithms can thenceforth be designed to be less reliant of the need for the extraction of the correct object topology. Some of the noteworthy examples are the papers by Murray, *et al.* [21], Lowe [17], and other approaches which expect only local cues such as the vertex-pair matcher of Thompson and Mundy [30]. These works have demonstrated that polyhedral descriptions can be extracted from scenes of varying complexity with acceptable degrees of success.

1.2 The need for indexing mechanisms in recognition

The algorithms given in [17, 21, 30] tend to lead to relatively inefficient processes for object recognition from (potentially) large model bases. This is because they do not make use of indexes to hypothesize object identities before attempting to solve the correspondence problem. Rather, they search through the model base sequentially testing out all models until a suitable match is found. A number of different algorithms have been offered in recent years which provide access to a model base through indexing rather than using model-image based search. A selection of the algorithms are those by Clemens and Jacobs [6], Forsyth, *et al.* [11], Lamdan and Wolfson [16], Califano and Mohan [3], and Stein and Medioni [27]. Following in their footsteps, we believe it prudent to produce indexing functions for polyhedra. Given such indexing functions, we can then opt for architectures which are likely to lead to efficient recognition algorithms for larger model bases of polyhedral objects.

Many examples of effective indexing functions are based on geometric invariants. We have therefore decided to base our investigation on invariant shape measures. The precise recognition application we pursue influences our choice between using the invariants of different geometric groups (Euclidean, affine or projective); we have chosen to use projective measures. This is principally due to the result of Rothwell, *et al.* [24], who demonstrated that all of the possible reconstructions which can be recovered from a single image of a four degree of freedom (dof.) polyhedron are in the same projective equivalence class.[1] The

[1] We refer the reader to Sugihara's book [28] for the definition of *degrees of freedom* of a shape.

invariant shape measures presented in [24] were recovered from single views of a polyhedron by reconstructing in three-dimensional space the family of polyhedra matching the image observations, and then taking characteristic measures of that family. As all of the members of the family are projective versions of each other (they all lie in the same *projective equivalence class*), they all have equal projective invariants.

The description in [24] is a generalisation of the approach of Sugihara [28] to allow the freedom to use un-calibrated cameras, and it also demonstrates that the recovered polyhedral descriptions all lie within the same projective equivalence class. Thus one actually retrieves a unique projective description of the imaged polyhedra rather than a family of Euclidean descriptions such as for [28]. Sugihara previously needed to use other shape cues based on texture or shading to reduce the ambiguity in the description to one which would be applicable as an index for a model base.

Nevertheless, the applicability of the method of Rothwell, *et al.* was limited and referred only to four dof. objects. We complete their theory within the domain of trihedral objects by writing out all of the invariants which are measurable for arbitrary dof. polyhedra. This generality was lacking in the original presentation, and so we complete the description paradigm for trihedral polyhedra.

1.3 Back to real images

Being able to produce invariant descriptions which facilitate indexing does not mean that the problems associated with polyhedral recognition have gone away. In all, we are not really any closer to being able to recognise polyhedra in real scenes (regardless of the size of model bases to which we make reference). The significant problem of extracting the structure of the two-dimensional projections of polyhedra still remains. In our opinion (which is certainly not exclusive), there is still a significant amount of work which needs to be done towards resolving problems in this domain. However, as the existence of the few working polyhedral recognition systems bears testament, at least minimal polyhedral descriptions can be extracted from images in certain limited environments.

One approach gaining in popularity is the use of unconnected data sets for the basis of structure recovery. Most of these have invested interest in the use of points and lines, and so algorithms have been liberated from having to reason about junctions and topology in images. Typical examples of this type of reasoning are the algorithms presented by Beardsley, *et al.* [2], Deriche, *et al.* [8], Harris [13], and Mohr, *et al.* [19]. Yet, one is still left with the task of having to reason about more global structure (such as topology) in three-dimensional space once the geometric algorithms have been applied. This is itself only a way to delay discussion of the problem, and so resolves little. The approach presented in this report allows us to take an alternative route. This route provides a compromise between the belief that no topological structure is available, and that which envisages the extraction of the complete topology of scene projections. Needless to say, neither of these standpoints make optimal use of the available information as there is some topology present, but it is frequently fragmented.

1.3.1 Local shape description

We make a return to the local hypothesis reasoning presented by Ettinger [10] for features of the curvature primal sketch [1], and generalized for invariant feature sets by Rothwell [25]. Our goal is always to measure local shape descriptions which are sufficiently simple to be robust, and then we exploit *joint-invariants* between the local descriptions to build up a more complete hierarchy which bears a resemblance to global object shape. On the theoretical level we achieve the part-whole decomposition by drawing the invariant description of [24], and the invariants based on the *butterfly configuration* of Sparr [20] and Mundy [32] together. The butterfly invariant is a geometric description of a special six-point configuration which we pursue in greater detail in a subsequent section in this paper.

Our interest in the butterfly invariant was promoted by the recent paper of Sugimoto [29]. This paper discusses an invariant very similar to the butterfly invariant, but suggests an algebraic rather than a geometric formulation. However, the key point is that Sugimoto suggested that the invariants in [29] in some way replace the invariants described by [24]. In fact, these two types of invariant can be taken hand-in-hand and are exactly complementary. In this paper we demonstrate that the original polyhedral invariant can be expressed fully in terms of the butterfly configuration. This relationship does not render the polyhedral invariant obsolete as the butterfly approach fails to provide a complete description of the global geometry of general polyhedra, and so it cannot be used to reason about the numbers and types of invariants which we expect to recover (whereas the original formulation does).

The derivation presented by Sugimoto is also somewhat awkward, and fails to give deep theoretical insight into the reasons for the existence of the butterfly invariant. We present a simpler algebraic formulation for the invariant which caters for the geometry in the same way as Sugimoto's invariant, but whose form provides more understanding. From this form we can quickly realise that the 3D projective invariant is actually just a ratio of affine invariants. This structure suggests familiarity with the planar projective invariants which we have seen before, for instance in Springer [26].

Through reasoning about the butterfly invariant we process polyhedra by breaking them down into local parts, each subpart being a butterfly. The process can be applied readily to polyhedra which have four dof. In the simplest case these shapes are composed of six planes. It is then straightforward to show that these four dof. figures are actually the building-blocks of more complicated structures in that higher dof. figures are simply composed of sets of articulated four dof. figures which may be treated somewhat independently, but are also tied together by families of planar projective invariants.

1.4 Summary

In summary, the scientific contributions of this paper are three-fold:

1. In Section 2 we demonstrate how the original invariant description of [24] can be decomposed into a set of three independent butterfly invariants. We also provide a novel algebraic form of the butterfly invariant. Prior to Sugimoto [29], the butterfly had been expressed geometrically.

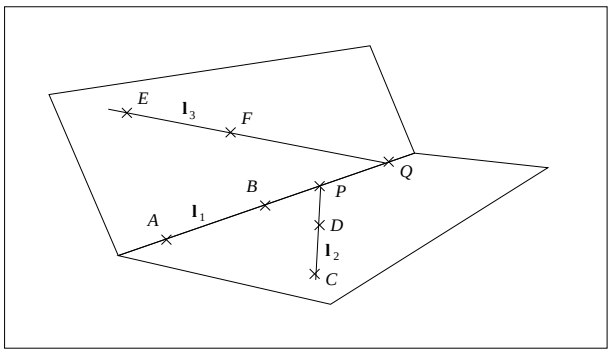


Figure 1: *The butterfly configuration consists of two sets of four coplanar points. Two points are shared between the sets. From this point configuration we can construct a cross ratio and hence a projective geometric description.*

2. We then show in Section 3 how to reduce a five dof. figure into sets of figures with four dof. For the simplest case of such an object (conceptually two cube-like structures stuck together using a common plane), we can recover three invariants for each half of the object, and a further eight joint-invariants based on planar measures between the two halves. We also describe how higher dof. polyhedra can be broken down into four dof. objects with pairwise planar constraints.

3. In Section 4 we report on algorithms for the extraction of the polyhedral descriptions and their invariants from real images. This section is perhaps the most important for recognition system builders as it demonstrates how correct topological descriptions can in fact be recovered from images when suitable care is taken during early visual processing.

2 The butterfly invariant

The butterfly invariant is the simplest known invariant for a set of points in three-dimensional space. The point configuration is such that the six points in space are broken up into two four point groups, $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ and $\{\mathbf{A},\mathbf{B},\mathbf{E},\mathbf{F}\}$. The four points in each group are coplanar, and two points are shared between the groups. The configuration is shown in Fig. 1.

The invariant for the butterfly is measured through the construction of a cross ratio. As can be seen in Fig. 1, it is possible to form a set of four collinear points and hence the invariant cross ratio [9]. These are the points $\{\mathbf{A},\mathbf{B},\mathbf{P},\mathbf{Q}\}$ from which we define the cross ratio $\tau = \{\mathbf{A},\mathbf{B};\mathbf{P},\mathbf{Q}\}$. The points must be collinear because they lie on the intersection line $\mathbf{l}_1$ of the two planes. In fact, $\mathbf{P}$ is found by intersecting $\mathbf{l}_1$ and $\mathbf{l}_2$ in the image, and $\mathbf{Q}$ from $\mathbf{l}_1$ and $\mathbf{l}_3$.

It is conceptually easy to compute this invariant in an image given the observations of the points: simply construct the relevant lines passing through the points, intersect the lines to give the necessary points, and substitute them into an expression based on the distances between the points:

$$\tau \quad = \quad \frac{AP}{AQ} \div \frac{BP}{BQ}. \tag{1}$$

It is clear that this value is truly invariant (and in fact it is the only invariant which exists for such a configuration) because the imaging process does not affect incidence in three-dimensional space and hence the construction of the points $\mathbf{P}$ and $\mathbf{Q}$. Proof of invariance then rests only on the assurance that the cross ratio is projectively invariant (Duda and Hart [9]).

A similar invariant for a set of five lines has been suggested by Sugimoto [29]. Two of the lines lie on one plane, two on an other, and the fifth line is the intersection line of the two planes (and is thus analogous to $\mathbf{l}_1$ in Fig. 1). By intersecting each of the first four lines with the fifth we observe a set of four collinear points and hence a cross ratio. However, Sugimoto proved the existence of the invariant using algebraic manipulations on sets of planes defining the lines in 3D. This process, although complicated, led to an invariant based on the determinants of 3×3 matrices which are defined by the coordinates of the lines in the images.

We now proceed to show that the geometric form of the butterfly can be turned *very simply* into an algebraic similar to Sugimoto's. The simplicity of the argument is its strength. Using an algebraic form for the butterfly firstly leads towards a more efficient implementation than the geometric calculation, but also interesting actually introduces us to a new *geometric* interpretation for the butterfly invariant.

2.1 The algebraic form of the butterfly

A homogeneous plane point is expressed by the three-vector $\mathbf{x} = (x, y, z)^\top$. Now, referring to Fig. 1 we need first to define $\mathbf{l}_1$ and $\mathbf{l}_2$. As is common practice, we represent a line satisfying $ax + by + c = 0$ by the vector $\mathbf{l} = (a, b, c)^\top$. Subsequently, the point-line constraint is expressed more efficiently as $\mathbf{l}.\mathbf{x} = 0$ (the '.' denotes the dot-product). The line passing through two points is expressed as the cross-product of the homogeneous forms of the points, and so $\mathbf{l}_1 = \mathbf{A} \times \mathbf{B}$ and $\mathbf{l}_2 = \mathbf{C} \times \mathbf{D}$.

Extending the process further we make use of the fact that the point defined by the intersection of two lines is given by the cross-product of the lines. Thus the point $\mathbf{P}$ is:

$$\mathbf{P} \quad = \quad \mathbf{l}_1 \times \mathbf{l}_2 = (\mathbf{A} \times \mathbf{B}) \times (\mathbf{C} \times \mathbf{D}),$$

Similarly we define $\mathbf{Q}$:

$$\mathbf{Q} \quad = \quad \mathbf{l}_1 \times \mathbf{l}_3 = (\mathbf{A} \times \mathbf{B}) \times (\mathbf{E} \times \mathbf{F}).$$

We may now substitute (algebraically) the coordinates of each of the six points into the expressions to find the coordinates of **P** and **Q** (these must be be normalised to yield the Euclidean coordinates in the image plane). These coordinates allow us to express algebraically the distances between the observed points **A** and **B**, and the estimated points **P** and **Q**. They can simply be substituted into eqn (1) to yield an expression the butterfly invariant. Then, by suitable algebraic manipulations, we are able to come up with the wonderfully simple result:

$$\tau \quad = \quad \frac{|\mathtt{M}_{ACD}|}{|\mathtt{M}_{AEF}|}.\frac{|\mathtt{M}_{BEF}|}{|\mathtt{M}_{BCD}|},$$

where $\mathtt{M}_{ijk}$ is the 3×3 matrix whose columns are the points **i**, **j** and **k** and $|\mathtt{M}|$ is the determinant of $\mathtt{M}$. The final result is that this very straightforward geometric and algebraic process leads us to a simple algebraic form for the butterfly invariant. Although the final result is actually very similar to that of Sugimoto (the difference being that there ratios of determinants were formed from the line parameters), the analysis is far less difficult.

2.2 What the algebraic form tells us

The algebraic form for the butterfly invariant leads to a very simple computational mechanism. However, it also provides significant insight into the underlying nature of this type of invariant. It immediately reminds us of a number of other invariants which can be expressed as ratios of determinants [26]. 3×3 determinants can be conceived of as being ratios of scaled areas. For instance, $|\mathtt{M}_{ACD}|$ is proportional to the area of the triangle ACD. Thus, τ is quite simply a ratio of ratio of areas; the areas being defined by triples of points on the two planes. This also means that it is a ratio of the two *affine invariants* which act on the different planes (these invariants can be defined as $|\mathtt{M}_{ACD}|/|\mathtt{M}_{AEF}|$ and $|\mathtt{M}_{BCD}|/|\mathtt{M}_{BEF}|$).

2.3 Computing polyhedral invariants with the butterfly

The projective invariants for polyhedral figures can be computed straightforwardly using butterfly invariants. Here we show how to compute the invariants for a cube-like structure such as that shown in Fig. 2. In fact, we shall see later on that understanding the invariants for this type of figure is as far as we need to go to comprehend the invariants for all tri-hedral polyhedra. For a cube-like polyhedron one can measure three independent invariants. This is because the figure is defined by six independent planes, and so has $6 \times 3 = 18$ dof. As the three-dimensional projective group has 15 parameters, there are $18 - 15 = 3$ independent invariants which can be measured (see [11] for an explanation of this counting argument). Three such independent butterfly invariants are:

$$\tau_1 = \frac{|\mathtt{M}_{ACE}|}{|\mathtt{M}_{ADG}|}.\frac{|\mathtt{M}_{BDG}|}{|\mathtt{M}_{BCE}|}, \qquad \tau_2 = \frac{|\mathtt{M}_{ADF}|}{|\mathtt{M}_{ABE}|}.\frac{|\mathtt{M}_{CBE}|}{|\mathtt{M}_{CDF}|}, \qquad \tau_3 = \frac{|\mathtt{M}_{ABG}|}{|\mathtt{M}_{ACF}|}.\frac{|\mathtt{M}_{DCF}|}{|\mathtt{M}_{DBG}|}. \tag{2}$$

It is a simple exercise to demonstrate that these three invariants are independent and form a basis for all of the other possible invariants.

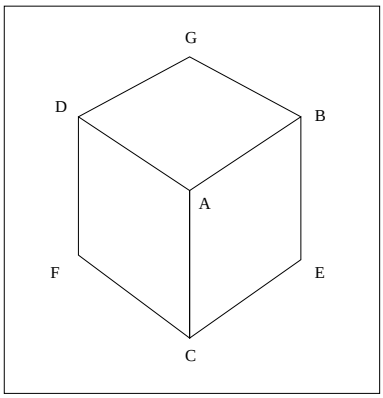


Figure 2: *The labelling of a cube used as a basis of the butterfly invariant computation.*

2.4 Equivalence to the original polyhedral invariants

Proof that the invariants given in eqn (2) are actually independent and as such form a basis for all other computable invariants depends on their exact equivalence to the three invariants given in [24]. It is a relatively simple algebraic task to prove that the three latter invariants are independent, and due to the counting argument of [11] it is known that there must be exactly three independent invariants for this configuration. Thus, once equivalence of the two types of invariants has been proved, we are free to draw conclusions about independence and completeness of the basis of the butterfly invariants.

The invariants given in [24] are exactly equivalent to the three-dimensional structure of the object under view. This is because the reconstruction process used in deriving the invariants computes the full projective shape of the imaged polyhedra. The invariant set provides the reconstructed coordinates of the sixth plane of the polyhedron within a frame defined by the first five. However, the process for demonstrating that the same is true for the butterfly form of the polyhedral invariants is also relatively simple. In short, they likewise enable the complete three-dimensional reconstruction of any imaged polyhedron. Consequently, the old and new forms of the invariants are equivalent.

We prove that reconstruction is possible using Fig. 3. Within three-dimensional projective space, which is where the polyhedron shown is embedded, we are free to fix fifteen projective parameters that define the projective representation. Therefore, we start off by fixing fifteen of the parameters of figure in three-dimensional space. We are further supplied with three constraints by the cross ratios, and so have eighteen in all with which to define the figure. Thus, to start with, we place the four points $\{A, B, C, D\}$ at the origin and at unit points on the x, y, and z axes. These points are marked in the figure and they use up $4 \times 3 = 12$ of the 15 dof. of the projective representation. We then use up the final three by placing the construction points $\{E, G, I\}$ on the extensions of the polyhedral edges as shown: each of these use up only a single projective dof. as they are constrained to lie on specific lines.

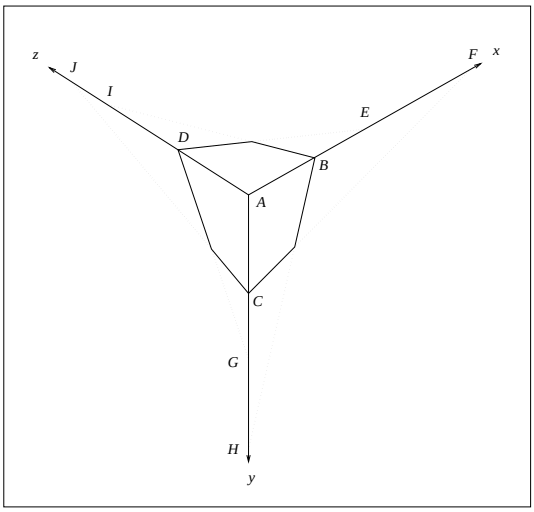


Figure 3: *From the three cross ratio invariants we can reconstruct the complete six-sided polyhedron. Details of the process are given in the text.*

The placing of these points is *absolutely arbitrary* so long as none of them coincide with $\{A, B, C, D\}$. Then, given the locations of $\{A, B, E\}$, and the first cross ratio τ_1, we can estimate the position of F uniquely. Points H and J are defined similarly by τ_2 and τ_3. The rest of the polyhedral construction can be completed in three-dimensional space as suggested by the figure.

3 Simplifying higher order polyhedra

In this section we extend previous theories on trihedral polyhedra to the general case. The results we give are significant because they resolve a number of important but unanswered questions which were posed by Rothwell, *et al.* in [24]. There, it was shown that invariants can be computed for polyhedral figures which have four dof. These represent only a relatively simple class of polyhedra such as cubes, as well shapes equivalent to cubes but with simple volumes cut-out of them. Some trihedral figures which have four dof. are shown in Fig. 4. The number of dof. of a polyhedra represent the dimensionality of the space of reconstructions which arise out of the image constraints.

In Section 2 we suggested a practical way to compute the invariants for four dof. objects. The motivation for extending the underlying trihedral theory is so that we can substantially increase the applicability of the polyhedral invariants to more general objects and so make use of the results within, for example, recognition applications. In the following paragraphs we demonstrate that figures with more than four dof. can be decomposed into sets of poly-

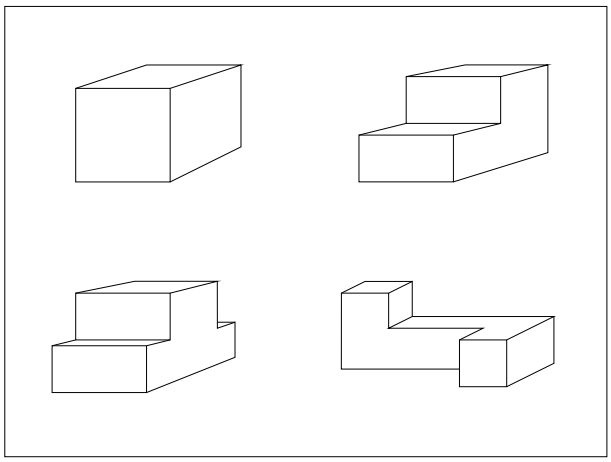


Figure 4: *Some four dof. figures. The most common example is a cube.*

hedra which have only four dof. The simplest example we consider is the decomposition of a specific form of five dof. figure into a pair of four dof. figures; this is shown in Fig. 5. More complicated figures can be decomposed similarly. The simplification process means that the invariant computations we have seen previously for four dof. polyhedra (such as the butterfly invariants) can be employed for each subpart of a more complicated structure, and so informs us that the basic invariants which can be computed for a general polyhedral structure are based on those for the separate four dof. figures. However, no projective invariants of the complete structures exist as each 4 dof. subpart maintains a certain amount of (non-projective) independence from the other parts. We prove this claim in Section 3.3.3.

Additionally, we can compute joint-invariants between all of the pairs of four dof. figures within the subpart hierarchy; these joint-invariants provide the glue which holds the subparts together. Therefore, although there are no *general* projective constraints between the subparts of a polyhedron, they are not reconstructed in space with total liberty, but are placed in related frames. Usually, each related pair of subparts in a figure share a common plane in which *plane projective invariants* enforce constraints. In Fig. 5 the plane is marked as the plane π.

These two sets of invariants represent all of the invariants which may be computed for an arbitrary trihedral polyhedra. We now proceed with proofs and explications of the statements given above. Due to limitations of space we are unable to repeat the theory given in [24], but only highlight the key points. However, we do make use of the same notation.

3.1 Reconstructing polyhedral figures

The assumptions we make about the polyhedra we treat are:

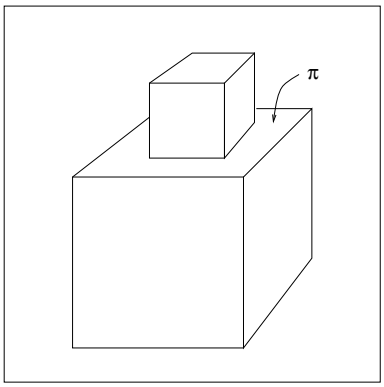


Figure 5: *A five dof. such as this can be split into two four dof. subpart which share a common plane π. In this case the polyhedron is simply one cube sitting on one plane of another cube. Each of the four dof. figures are obviously a cubes.*

1. They are made up of trihedral vertices (no more than three planes can meet at a point). This satisfies a genericity condition on the figures;
2. The polyhedra represent bounded solids which, due to the inherent properties of self-occlusions in imaging, we might as well consider *not* to have interior volumes and surfaces.

A direct result of these two facts is that any *complete set* of adjacent planes forms a solid polyhedra. A complete set means that three planes meet at every vertex, and two planes at every edge. By exploiting the trihedral constraint we can always render a set of adjacent planes complete by defining additional planes passing through edges and vertices.

We use a 3D coordinate system (X, Y, Z) and so planes satisfy:

$$a_j X_i + b_j Y_i + c_j Z_i + 1 = 0, \tag{3}$$

$j \in \{1, ..., n\}$ where n is the number of planes on the object. Each of the (X_i, Y_i, Z_i) represent the polyhedral vertices. Under the pinhole projection model, projection onto the plane $Z = 1$ maps the point (X_i, Y_i, Z_i) to (x_i, y_i) where: $x_i = \frac{X_i}{Z_i}$ and $y_i = \frac{Y_i}{Z_i}$. Substituting these into eqn 3, dividing through by Z_i and setting $t_i = 1/Z_i$ yields:

$$a_j x_i + b_j y_i + c_j + t_i = 0. \tag{4}$$

As stated in [24], this equation is applicable even when the calibration is unknown.

From the trihedral assumption we know that each image point lies on three planes, say $j \in \{p, q, r\}$. We eliminate t_i in eqn (4) between pairs of equations to give two linear equations per image point:

$$\begin{aligned} (a_i - a_j)x_p + (b_i - b_j)y_p + (c_i - c_j) &= 0, \\ (a_j - a_k)x_p + (b_j - b_k)y_p + (c_j - c_k) &= 0, \end{aligned} \tag{5}$$

and hence from the m observed image vertices we form a set of $2m$ equations in the $3n$ unknowns:

$$\mathtt{A}(\mathbf{x},\mathbf{y})\ \mathbf{w} = 0, \tag{6}$$

where $\mathbf{x} = (x_1, \ldots, x_m)^\top$, $\mathbf{y} = (y_1, \ldots, y_m)^\top$, $\mathtt{A}(\mathbf{x},\mathbf{y})$ is a $2m \times 3n$ array of constraints, and $\mathbf{w} = (a_1, b_1, c_1, a_2, \ldots, c_n)^\top$. We may now state the following theorem:

Theorem 3.1 *The kernel of the matrix* A *represents all of the possible solutions to the reconstruction of a trihedral polyhedra imaged in a single view by a pinhole camera.*

Proof The proof is in the derivation above, and more completely in [24]. □

3.1.1 Representing the solution as a set of vectors

We may represent the kernel of A as:

$$\mathbf{w} \quad = \quad \sum_{i=1}^{d} \lambda_i \mathbf{b}_i.$$

As is noted in [24], we can orient the basis for $\mathbf{w}$ so that:

$$\begin{aligned} \mathbf{b}_1 &= (1,0,0,1,0,0,\ldots,1,0,0)^\top, \\ \mathbf{b}_2 &= (0,1,0,0,1,0,\ldots,0,1,0)^\top, \\ \mathbf{b}_3 &= (0,0,1,0,0,1,\ldots,0,0,1)^\top, \end{aligned} \tag{7}$$

We may also fix the other basis vectors so that:

$$\mathbf{b}_i = \quad (0,0,0,(\mathbf{r}_i)_2^\top,\ldots,(\mathbf{r}_i)_n^\top)^\top,$$

for $4 \leq i \leq d$. This means that the solution for $\mathbf{w}$ with $\lambda_i = 0$ for $i > 3$, represents a simple planar configuration (strictly at least one of λ_i should satisfy $\lambda_i \neq 0$ for $i \leq 3$). The initial three elements of $\mathbf{b}_i$ being zero means that the first plane has coordinates $(\lambda_1, \lambda_2, \lambda_3, 1)^\top$.

3.2 Four dof. polyhedron

It was shown in [24] that for $d = 4$ the solutions represented by $\mathbf{w}$ are all projectively equivalent to each other. In this case we thus need consider only one solution from $\mathbf{w}$ for the computation of the three-dimensional projective invariants on the shape. The solution chosen usually was that with $\lambda_i = 0$ for $i \leq 3$, and $\lambda_4 = 1$. Once computed, the invariants can be used for indexing into a model library, or simply as a compact representation of the three-dimensional shape.

3.3 Five dof. polyhedron

Consider now the case for $d = 5$. *This has not been treated previously.* To simplify the discussion we represent $\mathbf{b}_4$ and $\mathbf{b}_5$ by:

$$\mathbf{b}_4 = (0, 0, 0, \mathbf{r}_2^\top, \ldots, \mathbf{r}_n^\top)^\top \qquad \text{and} \qquad \mathbf{b}_5 = (0, 0, 0, \mathbf{s}_2^\top, \ldots, \mathbf{s}_n^\top)^\top.$$

Note that the first plane has $\mathbf{r}_1$ and $\mathbf{s}_1$ equal to the null vector.

3.3.1 Separating out the first four dof. figure

We return to the form of the constraints given in eqn (5). Consider the first plane, and another plane i which shares an edge with it; the endpoints of the edge are the points p and q. First we construct a set of planes $\mathcal{S}_1$ which initially contains the first and the i^{th} planes. Then, from the plane-point constraint of eqn (5) we know that:

$$\begin{aligned} (a_i - a_1)x_p + (b_i - b_1)y_p + (c_i - c_1) &= 0, \\ (a_i - a_1)x_q + (b_i - b_1)y_q + (c_i - c_1) &= 0. \end{aligned} \tag{8}$$

Consider writing $\alpha = (\lambda_1, \lambda_2, \lambda_3)^\top$. Then, making use of the fact that both $\mathbf{r}_1$ are $\mathbf{s}_1$ are null vectors means that $(a_1, b_1, c_1)^\top = \alpha$ and $(a_i, b_i, c_i)^\top = \alpha + \lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i$.

Subsequently, eqn (8) reduces to the following vectorial constraints:

$$\begin{aligned} (x_p \; y_p \; 1)^\top \,.\, (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i) &= 0, \\ (x_q \; y_q \; 1)^\top \,.\, (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i) &= 0, \end{aligned} \tag{9}$$

These define $(\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i)$ uniquely up to a scale as $\mathbf{x}_p$ and $\mathbf{x}_q$ must be distinct points (two linear constraints fix three homogeneous unknowns). This solution exists for all choices of λ_4 and λ_5, and so we deduce that either:

1. $\mathbf{r}_i$ is parallel to $\mathbf{s}_i$;
2. Either $\mathbf{r}_i$ or $\mathbf{s}_i$ is the null vector (but not both if the first and i$^{\text{th}}$ planes are distinct).

Within the bounds of the geometric constraints used to compute the basis vectors $\mathbf{w}$, we are free to orient both $\mathbf{b}_4$ and $\mathbf{b}_5$ as we desire. In order to simplify the overall description, we now make a further change of basis. If we make the assumption that $\mathbf{r}_i$ and $\mathbf{s}_i$ are parallel, a change of basis is effected so that $\mathbf{s}_i$ becomes a null vector, and $\mathbf{r}_i$ is non-zero. Should either of the two vectors have already been null we can make a suitable change of basis (perhaps trivially) to ensure that $\mathbf{r}_i$ is non-zero and $\mathbf{s}_i$ is null. Having made this change, it is clear that neither the first plane, nor plane i, depend on λ_5. By reapplying the reasoning, we may algebraically extend the process and provide independence to λ_5 for all of the other planes which happen to have $\mathbf{r}_j$ parallel to $\mathbf{s}_j$ (or are null vectors). All planes of this type are included in the set $\mathcal{S}_1$.

It is certain that a number of planes will be included in $\mathcal{S}_1$. Geometrically we see that providing independence to λ_5 for other planes is based on the trihedral assumption underlying our manipulations. Consider any third plane which shares an edge with both the first and i^{th} planes; the third plane is precisely defined by the positions of the other two. The first plane is independent of both λ_4 and λ_5, and the i^{th} one can depend only on λ_4. Thus it is clear that the new plane depends only on λ_4 (and so its $\mathbf{s}$ must be null). This process can be continued for all other planes which have two edges in common with a pair of planes from $\mathcal{S}_1$. One will thus build up a set of planes constituting $\mathcal{S}_1$ which depend only on λ_4 and that consequently represent a four dof. figure.

3.3.2 The second four dof. figure

Once the first four dof. subpart has been extracted we can be sure that no planes remain outside $\mathcal{S}_1$ which are in contact with more than one plane in $\mathcal{S}_1$. This is quite simply because if a plane not in $\mathcal{S}_1$ were attached to two planes in $\mathcal{S}_1$, then it would have to be fixed in space. We label the complementary set to $\mathcal{S}_1$ by $\mathcal{S}$. We may now make the following statements which have straightforward justifications:

1. The planes in $\mathcal{S}_1$ use up four of the five dof. for the polyhedron. This is simply because the planes in $\mathcal{S}_1$ are defined by λ_i, $i \leq 4$.

2. $\mathcal{S}_1$ does not represent the entire polyhedra ($\mathcal{S}$ is not empty). If it did, then $\mathbf{b}_5$ would be null and we would have only a four dof. figure.

3. There must be at least one plane in $\mathcal{S}$ which is adjacent to a plane in $\mathcal{S}_1$. If this were not the case, then the planes in $\mathcal{S}$ would make up a figure with at least three more dof. These extra three dof. would lead to the entire figure having too many dof.

We exploit the third of these statements to base the rest of the reconstruction around the plane common to $\mathcal{S}_1$ and $\mathcal{S}$. In order to achieve our goal, we must yet again change the basis so that the common plane is accounted for by the first three parameters of $\mathbf{w}$. For simplicity of argument we also reorder the rest of planes so that those dependent only on λ_4 appear first in the basis vectors. The m^{th} plane (which is the first plane that does not belong to $\mathcal{S}_1$) is also chosen so that is shares an edge with the first plane. The following form of basis vectors is therefore produced:

$$\begin{aligned} \mathbf{b}_4 &= (0,0,0,\mathbf{r}_2^\top,\ldots,\mathbf{r}_{m-1}^\top,\mathbf{r}_m^\top,\ldots,\mathbf{r}_n^\top)^\top, \\ \mathbf{b}_5 &= (0,0,0,\ldots\ldots,0,0,0,\mathbf{s}_m^\top,\ldots,\mathbf{s}_n^\top)^\top. \end{aligned}$$

We now simply repeat the process which we used previously for the elimination of the components $\mathbf{s}_i$ from $\mathbf{b}_5$ for $i < m$, though this time we eliminate the $\mathbf{r}_i$. Given that the first plane is represented by the zero vector, and that the m^{th} plane shares an edge with it, we know that $\mathbf{r}_m$ and $\mathbf{s}_m$ are parallel (cf. eqn (9)). Now, we may again eliminate either of $\mathbf{r}_m$ or $\mathbf{s}_m$ to yield dependence only on one of λ_4 or λ_5. Clearly this plane cannot depend on λ_4

otherwise it would already have been included within $\mathcal{S}_1$. Therefore, we are able to eliminate $\mathbf{r}_m$. In the same way that we added planes to $\mathcal{S}_1$ we continue to build up planes dependent only on λ_5. We place all of these planes in the set $\mathcal{S}_2$.

Once all possible $\mathbf{r}_i$ have been set to zero, we are left with basis vectors of the form:

$$\begin{aligned} \mathbf{b}_4 &= (0,0,0,\mathbf{r}_2^\top,\ldots,\mathbf{r}_{m-1}^\top,0,0,0,\ldots,0,0,0,\mathbf{r}_k^\top,\ldots,\mathbf{r}_n^\top)^\top, \\ \mathbf{b}_5 &= (0,0,0,\ldots\ldots,0,0,0,\mathbf{s}_m^\top,\ldots\ldots,\mathbf{s}_{k-1}^\top,\mathbf{s}_k^\top,\ldots,\mathbf{s}_n^\top)^\top. \end{aligned} \tag{10}$$

Such a basis represents a decomposition of the polyhedron into two four dof. figures built around the first plane (the first one of planes 1 to $(m-1)$, and the second one plane 1, and planes m to $(k-1)$). The two four dof. subparts are both complete solid polyhedra. In addition to these sub-figures, there also potentially exists a set of planes k to n which have yet to be considered. These are represented by the set $\mathcal{S}_3$. For the polyhedron in Fig. 5 it is trivial to see that $\mathcal{S}_3$ is empty, and so $k = n+1$.

However, this is not the general case and $\mathcal{S}_3$ might well contain further polyhedral subparts. Note that the planes in this subsidiary set are still constrained completely by λ_4 and λ_5. $\mathcal{S}_3$ thus has the appearance of being a more complex figure than the two four dof. subparts which we have already extracted, though this is not in fact the case. $\mathcal{S}_3$ will always be composed of only a set of four dof. polyhedra.

The simplest way to understand this is to reorder the planes so that one of the common planes between $\mathcal{S}_1$ and $\mathcal{S}_3$ is the first plane (such a plane must exist). We then repeat the above elimination process. Doing this would mean that a subset of planes in $\mathcal{S}_3$ would be dependent only on the new λ_5 ($\mathcal{S}_1$ is still parametrized by λ_4) and so would essentially have only four dof. By progressive choices of common planes between the extracted subparts of $\mathcal{S}_3$ and their complement in $\mathcal{S}_3$, we can gradually demonstrate that $\mathcal{S}_3$ is composed only of four dof. figures. The trick is to see that any two polyhedra sharing a common plane can always be demonstrated to have four dof. We then just need to consider all of the common planes.

An example of a figure with three different sub-polyhedra is shown in Fig. 6. Each time we use a different plane π_i, $i \in \{1,2,3\}$, as the first plane, we immediately recover two polyhedra as being dependent on either λ_4 and λ_5, and the other as depending on both. However, it is clear that all three subparts are in fact specially constrained four dof. polyhedra. Thus, we can extend our current understanding of trihedral polyhedra via the following claim:

Theorem 3.2 *Five dof. trihedral polyhedra are made up of simple building blocks consisting only of constrained four dof. polyhedra. As each subpart has only four dof., their own local equivalence classes are projective.*

Proof As discussed above. □

We now show that the global equivalence class for an entire polyhedron is larger than projective.

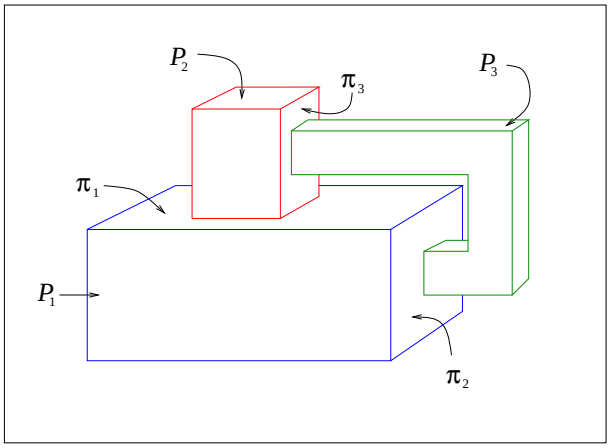


Figure 6: *A five dof. figure which is built around three different polyhedral subparts of four dof. The four parts are shown as P_i with common planes π_i, $i \in \{1, 2, 3\}$.*

3.3.3 Projective inequivalence of five dof. figures

Unlike the four dof. figures discussed previously in [24], two different reconstructions of a five dof. figure need not necessarily be projectively equivalent. We state this formally as follows:

Theorem 3.3 *The different reconstructions of a five dof. non-trivial polyhedron created from single views using an uncalibrated pinhole camera are not in the same projective equivalence class.*

Proof For the proof we need only consider the type of figure shown in Fig. 5. Such a figure represents the simplest possible case. All other cases reside within larger parameter spaces and so cannot have their reconstructions lying within single projective equivalence classes if the simpler figure does not.

A non-trivial polyhedron is one made up of at least one square face.[2] The only trivial polyhedra which exist with five dof. are composed of two tetrahedra sharing a plane. As these figures have five independent vertices (six vertices are coplanar, and so three of the coplanar vertices are dependent on the 6-3 others), they are obviously projectively equivalent.

The non-trivial case is more interesting. An uncalibrated pinhole camera leads to a reconstructed figure described by the forms of eqns (7) and (10). Given that at least one of the four dof. subparts has a face with four edges, it follows that it must also be composed of five planes. Let these five planes be defined by λ_4, which we fix arbitrarily. Five planes define a three-dimensional projective frame, and so λ_4 fixes our projective representation. However, leaving λ_4 fixed still allows us to alter λ_5 and hence move around the planes of

[2] Tetrahedra consist of only four points, and so yield no projective structure.

the second four dof. figure. The first five planes remain fixed, and so the projective frame remains unchanged, but the projective coordinates of a plane defined by λ_5 change with λ_5. Therefore figures with different choices of λ_5 are not projectively equivalent. □

3.4 Invariants for five dof. figures

There are two sets of invariants which can be measured for five dof. polyhedral figures. The first are based on the invariants of each of the subparts, and so are the same as the four dof. invariants given in [24]. The total number of invariants of this type depends upon the number of planes in each subpart. If the number of planes is denoted by n_i for each of the k subparts, then:

$$\sharp(I_1) = \sum_{i=1}^{k}(3n_i - 15).$$

For an object such as that shown in Fig. 5 there are six planes in each part ($k = 2$), and so a total of six invariants can be measured in this way.

We can also measure a number of invariants in the common plane. Again referring to Fig. 5, we see that there are eight edges lying in the common plane. Eight lines in a plane yield $8 \times 2 - 8 = 8$ projective invariants. Generically the number of invariants of this form depends on the precise structure of the polyhedra: if there are m_i edges in each of the j common planes between the subparts, $i \in \{1, \ldots, j\}$, then there are:

$$\sharp(I_1) = \sum_{i=1}^{j}(2m_i - 8)$$

computable planar invariants.

We now proceed to show that these represent all of the invariants which can be computed for such a figure.

3.5 Representing the degrees of freedom as a group

Theorem 3.4 *The equivalence class of the single polyhedral subpart controlled by the family of deformations parametrized by λ_5, with λ_4 fixed, is embedded within a four degree of freedom subgroup of the three-dimensional projective group. This four degree of freedom subgroup is the minimal enveloping linear group.*

Proof Define the first common plane to be the plane at infinity (or the only common plane if there is only one). This choice of frame is arbitrary, but simplifies the subsequent analysis. We are dealing with a four dof. subpart of a polyhedra, and so we know that all of the deformations possible are embedded within the 3D projective group. We need therefore only consider projective transformations. All of the linear actions which leave the plane at

infinity fixed (all points in the plane at infinity are stationary points) can be parametrized by[3]:

$$\mathtt{T} \;=\; \alpha \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & d \end{bmatrix}. \tag{11}$$

The camera must also remain fixed under this set of transformations. Furthermore, the camera cannot lie on the plane at infinity (otherwise we would not see it), and so we may place the camera centre at $\mathbf{C} = (X, Y, Z, 1)^\top$. So, as $\mathbf{C}$ must map to itself under $\mathtt{T}$, and eliminating the parameters $\{a, b, c\}$, we further constrain $\mathtt{T}$ by:

$$\mathtt{T} \;=\; \alpha \begin{bmatrix} 1 & 0 & 0 & Xd' \\ 0 & 1 & 0 & Yd' \\ 0 & 0 & 1 & Zd' \\ 0 & 0 & 0 & d'+1 \end{bmatrix}, \tag{12}$$

where we have written $d' = d - 1$. Now, as the position of the camera is unknown, we see that $\mathtt{T}$ still maintains four degrees of freedom (but they are now non-linear). Consequently, we might as well continue to use the linear form for $\mathtt{T}$ given in eqn (11) as this represents the smallest enveloping linear group for the form in eqn (12). □

Theorem 3.4 allows us to determine how many invariants can be measured for a polyhedral shape. Using the counting argument of [11] which states that the number of invariants is equal (when there is no isotropy) to the number of parameters of the algebraic system less the number of degrees of freedom of the group, we see that there should be $(6-1)\times 3-4 = 11$ invariants for the six-plane 4 dof. sub-polyhedra. The reason for there only being $(6-1)$ planes in the calculation is because one of the planes has been fixed as the plane at infinity, and so adds nothing to the projective structure.

In Section 3.4 we stated that the following invariants could be computed for a polyhedra such as the one shown in Fig. 5:

1. Three invariants for the six planes making up the figure;
2. Eight invariants for the edges of the second four dof. subpart in the common plane (these invariants are computed using the four edges in the common plane from the first four dof. subpart).

[3] The determinant of $\mathtt{T}$ must be non-zero, and so we are justified in placing unit values along the diagonal. Consequently, neither α nor d can be zero.

Therefore, a total of 11 invariants were considered. The theory in this section has demonstrated that a maximum of 11 invariants can be computed for such a figure, and so those given in Section 3.4 represent all of the invariants which can be computed.

The story is different for more complicated five dof. figures which have three or more subparts. For these, use of all of the four dof. invariants and all of the plane invariants provides over-constraint, and so it is in general difficult to predict *a priori* how many independent invariants can be measured. However, knowledge of the object's structure immediately indicates how many such invariants exist.

3.6 Higher degree of freedom figures

We can use the results of Section 3.4 to enhance our understanding of polyhedra with six or more dof. We proceed with higher dof. figures in a similar way to five dof. polyhedra. First, extract a four dof. subpart and define the planes which it has in common with the rest of the polyhedron. Then, continue to examine the rest of the polyhedron by ignoring the planes in the first subpart, extracting further four dof. polyhedra. In this manner the whole polyhedron can be decomposed into a set of four dof. subparts.

Justification that the process works for figures with more than five dof. arises through appeal to a generalization of eqn (9). Again we fix the first three components of the basis vectors of $\mathbf{w}$ so that $(a_1, b_1, c_1)^\top = (\lambda_1, \lambda_2, \lambda_3)^\top$. Then, considering the points $\mathbf{x}_p$ and $\mathbf{x}_q$ to be the endpoints of the edge defined by the intersections of the first and i^{th} planes, we can place the following constraints on the components of the higher order basis vectors (that $\mathbf{b}_i$, $i \in \{4, 5, 6, \ldots\}$):

$$\begin{aligned} (x_p \; y_p \; 1)^\top \,.\, (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i + \lambda_6 \mathbf{t}_i + \ldots) &= 0, \\ (x_q \; y_q \; 1)^\top \,.\, (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i + \lambda_6 \mathbf{t}_i + \ldots) &= 0. \end{aligned} \tag{13}$$

As we saw before, these two constraints define $(\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i + \lambda_6 \mathbf{t}_i + \ldots)$ uniquely *for all choices* of λ_i, $i \geq 4$. We thus know that all of the elements $\mathbf{r}_i$, $\mathbf{s}_i$, $\mathbf{t}_i$, etc., of the basis vectors are either parallel or zero. Consequently, we are led towards exactly the same reasoning mechanism which was used for five dof. figures in that we eliminate the $\mathbf{s}_i$, $\mathbf{t}_i$, etc., and leave the $\mathbf{r}_i$. This process again results in the extraction of an initial four dof. sub-figure dependent on λ_4.

The computation of the invariants for the entire object is then straightforward:

- Compute the projective invariants for the planes within each subpart.
- Compute the planar invariants between the edges in the different common planes defined by adjacent subparts.

It follows obviously from the result for five dof. polyhedra that the families of reconstructions for higher dof. figures are not projectively equivalent.

4 Finding polyhedra in images

In this section we report on a pair of recent approaches which show how polyhedral descriptions can be extracted from real images. As always, we are plagued by the difficulty of extracting accurate segmentation results from real images. We discuss two different methods and provide demonstrations of them working on relatively simple images containing well-defined polyhedral objects. The two solutions proposed are:

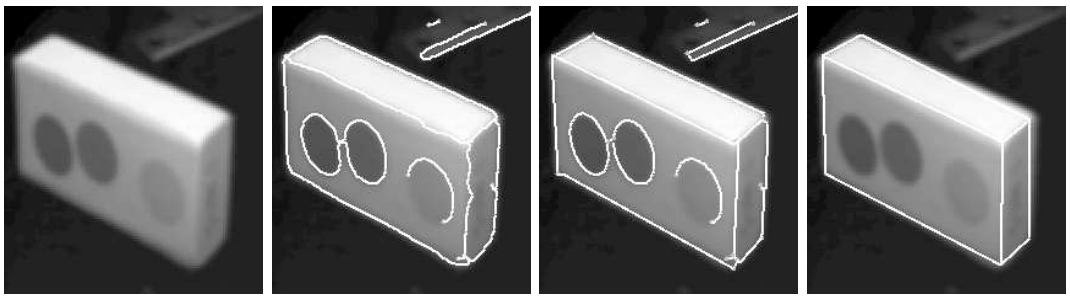
- We use an edge detector to initialize a polyhedral snake on sets of image features. The snake relies on the extraction of at least one image polygon, though in practice we find that the minimum number required to provide reliable constraint is either a pair of quadrilaterals or a single hexagon.
- We search for pairs of adjacent closed regions an image which can be used to estimate values for butterfly invariants. These invariants are used to index into a model base and subsequently to provide hypotheses suggesting which object might be present in the scene. Hypotheses are combined post-indexing to derive richer object descriptions.

Both of these approaches require the extraction of edges and lines in the image. The edge detector we use has certain similarities to the Canny [4] edge detector, though provides much better connectivity around junctions (by exploiting adaptive thresholding). Full details about the edge filter can be found in [23]. Both of the approaches we discuss rely on the extraction of good junction connectivity at the projections of a number of the polyhedral vertices. The fact that this is possible is demonstrated in the images. Straight lines are fitted to the edgel-chains using orthogonal regression and a topological (connectivity) structure composed of connected straight lines and edgel-chains segments is produced. Even though we are principally interested in the straight lines, the edgel-chain segments are maintained to provide a complete topological description in the image.

The actual data structures we use are built around a *vertex-edge-face* topology hierarchy. Vertices are typically used to represent junctions and the interface between pairs of lines or between a line and an edgel-chain segment. Edges link vertices and are thus represented geometrically by lines or edgel-chains. Faces are closed cycles of edges (we also have 1-chains which represent non-cyclic edge chains). This topology hierarchy allows the straightforward extraction of closed regions from the image as each *face* is a closed region.

4.1 Polyhedral snakes

The primary goal of the snake extraction approach is to recover structures which have the same topological forms as the projections of the polyhedra which we wish to recognize. Once these have been extracted, they can be matched to snake models, and then the snakes are allowed to develop in the image by interaction with the image intensity surface. After a number of iterations we can measure the invariants of the snake (which are similar to projected polyhedra), and thus hypothesize the identity of the object in the scene.



a **b** **c** **d**

Figure 7: *(a) shows a six-sided polyhedron which has an image width of about one hundred pixels. In (b), the output of the edge detector is superimposed; note that the basic topological structure of the edges is similar to that for a projected polyhedron. (c) shows the lines fitted to (b) using orthogonal regression (though the three circles are represented by circular arcs as the segmentation routines finds these to be a more suitable description). The mixed line and topological description is used to derive the polyhedral snake in (d).*

As is normal, the first phase of processing involves filtering the grey-scale image to find edges. The process is demonstrated in Fig. 7. Notice that we have been able to recover a fairly good description of the polyhedron's topology with the edge detector. Principally the polyhedron's edge description consists of three internal regions each of which match projected polyhedral faces. Being able to find these closed image regions means that we are able to reject scene clutter (non-polyhedral regions) relatively quickly during subsequent processing.

We then fit lines using orthogonal regression, and throughout we ensure the maintenance of the topological description originally recovered in the image. Single edge-chains may become fragmented into edgel-chain and line sequences connected by vertices. These vertices do not change the gross topological description in the image which is dominated by higher order junctions (for images of tri-hedral objects the important vertices are triple junctions). After fitting, we extract closed regions from the image. For Fig. 7 there are three basic regions which are suitable, plus a number of others which include combinations of these regions.

Subsequent analysis is focused on the regions and on the lines which they contain. These are shown in Fig. 7(c). Over-segmentation can arise due to the edge detector recovering small noisy features. A number of these are shown in Fig. 7(b). These features disrupt the line fitting process and cause breaks to be inserted between pairs of line segments. We therefore traverse the boundary of each region and test whether a pair of adjacent lines would be better represented by a single straight line segment. If so, a single line segment is substituted.

Once the line merging process is complete, we count the number of lines in each region to see whether it remains of interest. Polyhedral faces must contain at least three line segments, and so any with less can be discarded. For this application we desire at least four lines, and

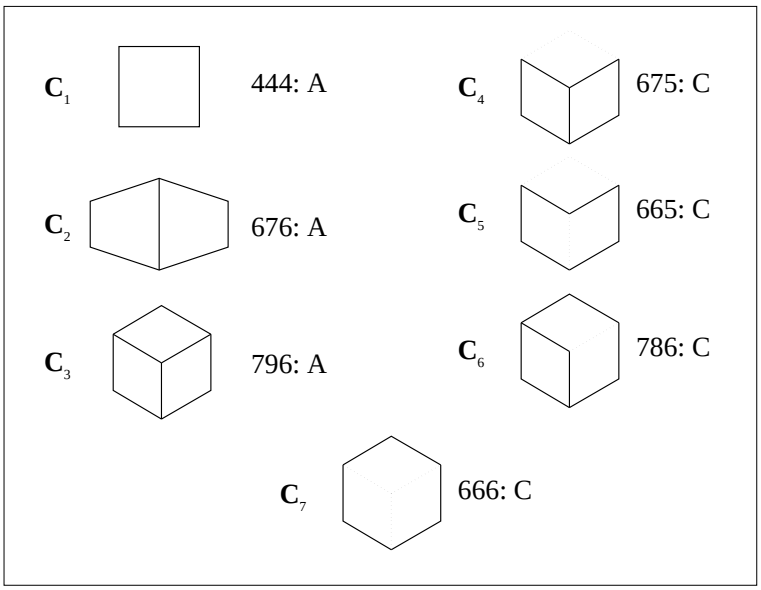


Figure 8: *The different topological descriptions we store for the projections of a six-plane polyhedron (such as a cube). Each of the seven types are characterized by a three digit code: the first digital gives the number of vertices in the description, the second the number of edges, and the third the number of vertices on the convex hull of the shape. Associated with each description is also a code, 'A' if the description is all that can be recovered from such a viewpoint, and 'C' if some edges are missing.*

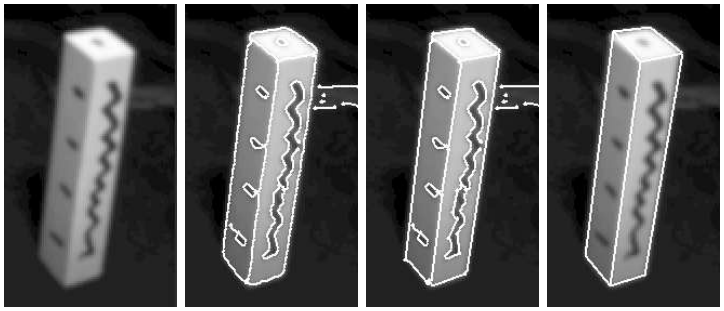
so this provides a yet tighter constraint. Consecutive line segment endpoints should also be reasonably close as polyhedral faces are generally polygonal. Given this, we can discard the edge-chains between lines and reduce the face representations to ones containing only polygons.

The penultimate state of processing involves the matching of the adjacency graph of the remaining regions to topological models which we have for the polyhedra in the model base. The description in Fig. 7(c) matches that superimposed in Fig. 7(d) (which has been deformed to fit the geometry). The complete set of descriptions from which we form snakes is given in Fig. 8.

Given a match, such as that shown, we initialise a snake which is allowed to relax onto the contrast boundaries in the original image. The snakes are a crude implementation of those described in detail by Kass, *et al.* [15], or Cipolla and Blake [5]. The final position of the snake yields sufficient structure for invariant computation. In Fig. 7, all of the butterfly invariants are unity because the polyhedron is projectively equivalent to a cube (which is

iteration	invariant 1	invariant 2	invariant 3
0	-0.920797	-0.897423	-1.34495
4	-0.956112	-1.02075	-1.04808
8	-0.964451	-1.01821	-1.02506

Table 1: *The three invariants for the polyhedral snake of Fig. 7 for the first eight iterations. The invariant values should be all be equal to unity.*



a **b** **c** **d**

Figure 9: *(a) shows the original polyhedron with superimposed edges in (b), fitted lines in (c), and polyhedral snake in (d).*

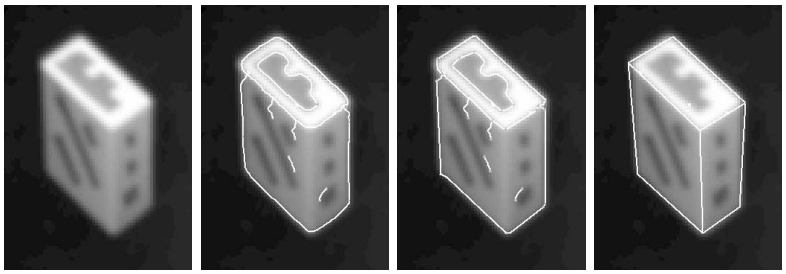
composed of a triple of parallel plane sets). Table 1 shows how the three invariants for the shape change over the iterations of the snake.

Another similar example of this type of processing is shown in Fig. 9. Again the edge detector is able to recover the complete topology of the projection of the polyhedron and so the same snake structure can be used as before. In Fig. 10 we have a harder case. The edge detector has failed to recover the vertical internal boundary, and so we must use a simplified model with which to initialize the snake. In this case we have a description equivalent to $\mathbf{C}_6$ (in Fig. 8) and so we are able to hypothesize the location of the missing edge by joining up a pair of vertices in the shape. We return to the complete snake description ($\mathbf{C}_3$) once the edge has been inserted. In some cases we have to estimate missing vertex positions by exploiting approximate parallelism; the action of the snake removes the effects of the approximation and usually yields suitable fits if the correct model hypothesis has been made.

In practice we have found that either two quadrilateral faces or a single hexagon are required to initialize the snake model. Should we find a lone quadrilateral, such as for $\mathbf{C}_1$, we do not continue processing.

4.2 Finding polyhedra using local butterfly invariants

The approach discussed above is global; we now describe a local method for solving the problem. The initial processing stages are the same in that we extract edges and lines, and



a **b** **c** **d**

Figure 10: *(a) shows the original polyhedron with superimposed edges in (b). Note that in this instance the complete topological structure of the polyhedron has not been recovered. Fitted are lines shown in (c), and the polyhedral snake in (d).*

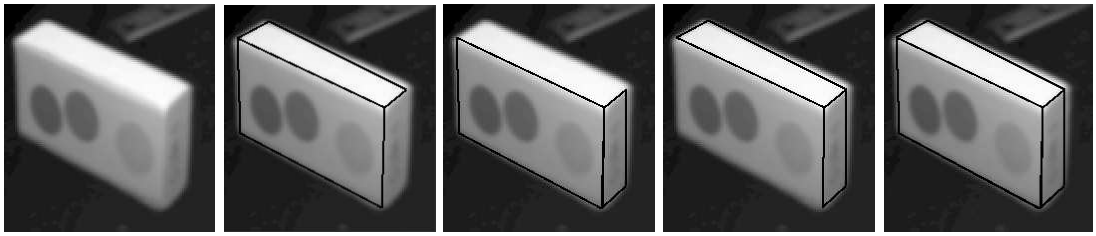
then recover *faces* from the image which represent closed regions. As before, these faces are processed to merge lines and are then represented as polygons if suitable. All polygons which are not quadrilateral are rejected as they are not appropriate for use with the butterfly invariant.

Faces which approximately (or exactly) share edges are then associated to produce a butterfly configuration. The vertices of the common edge are adjusted so that they lie suitably between the two faces (by averaging the image coordinates). From this merged configuration we can immediately compute a butterfly invariant. The combined face-pair feature is accepted depending on whether the invariant value belongs to a model in the model base (potentially this is evaluated through indexing). Each accepted invariant value can be used to form a local hypothesis. Larger hypotheses can be formed by combining the local hypotheses using the hypothesis extension process described in [25]. In brief, two hypotheses are consistent if they represent different parts of the same model, and if the feature correspondences between the image and the model are also consistent. The final results of the process (for the objects which we have so far examined), are the four dof. descriptions of the polyhedra in the images.[4]

In principal we can re-apply the hypothesis extension process between the four dof. figures by using the planar projective invariants in the common planes as tests of consistency. However, we have not yet been able to evaluate this procedure.

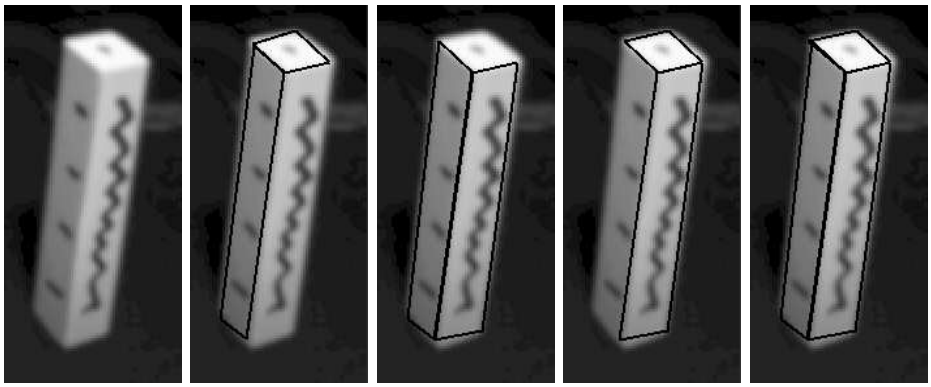
An example of the extraction of the butterflies is given in Fig. 11. In (b) to (d) we show the three different butterflies which were extracted for the image section shown in (a). These butterflies are formed by a total of three image faces each containing four principal line segments, and have invariant values of 1.007, 1.022, and 1.000 respectively. All of these invariants match butterfly invariants for objects which have projective equivalence to a cube.

[4]A corollary of the result given in Section 3 is that two faces sharing an edge must come from the same four dof. figure.



a **b** **c** **d** **e**

Figure 11: *After segmentation and closed region detection on the image section in (a) we can extract three butterflies with invariants approximately equal to unity. These butterflies can be merged into the single polyhedral description in (e).*

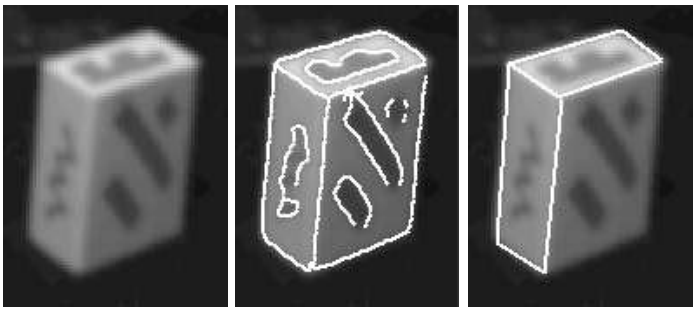


a **b** **c** **d** **e**

Figure 12: *We can recover three butterflies for this imaged polyhedra and hence derive the unified description in (e)*

They can thus be merged into the single polyhedral description shown in (e) which is a single four dof. polyhedra.

A second example of the measurement of the butterfly descriptions is given in Fig. 12. In this case the three butterflies have invariants equal to 1.000, 1.000, and 1.029. However, in Fig. 13 we show an example in which only a single butterfly can be recovered (with an invariant value of 1.000) due to the failure of the edge detector to recover more than two of the polyhedral faces as closed regions. In this case we have only weak support for a polyhedral hypothesis as one matching invariant carries less weight than three. Nevertheless, the precision of the single invariant suggests that the features probably do correspond to a polyhedron with similarity to a cube.



a **b** **c**

Figure 13: *For the polyhedron shown in (a) we can only recover two of the polyhedral faces as closed regions in the edge description of (b). Thus we can find only a single butterfly configuration (c). However, the invariant value for the butterfly matches precisely that which we would expect to find for such a polyhedron.*

5 Conclusions

We have made a number of different contributions in this paper. The first two are at a theoretical level, and the third expresses a pair of practical approaches for the extraction of polyhedral descriptions from images. The first theoretical study presented a very simple algebraic formulation for the butterfly invariant and showed how it is related to the polyhedral invariants of Rothwell, *et al.* [24]. The formulation of the invariant is both novel, and is far easier to understand than those presented previously.

The second main theoretical contribution is a completion of the projective invariant description of [24] for tri-hedral polyhedra to arbitrary degree of freedom objects (the original formulation dealt only with four dof. polyhedra). We demonstrate that all higher order polyhedra can be broken down into sets of four dof. polyhedra which are linked together via plane projective invariants. Although limitations of space have meant that we have been unable to prove all of the claims made within this article (though we have proved them elsewhere), we have in fact generated a complete description paradigm for arbitrary tri-hedral polyhedra.

Finally, we developed two different implementations of algorithms intended for the detection of polyhedra in real image. The first is founded on topological reasoning which is used as a basis for a polyhedral snake process. The second makes use of closed region detection for the creation of local hypotheses, and then the use of hypothesis extension for the generation of more complete polyhedral descriptions. The implementations are not yet fully mature, though we intend in the coming months to develop the ideas further and integrate both methods into a full object recognition system.

Acknowledgments

The authors would like to acknowledge the input of a number of people in this work. The original polyhedral work was based on discussions between CAR and David Forsyth; the work presented here is really only a completion of some of the early ideas. We are also grateful for discussions with Olivier Faugeras, Joe Mundy and Andrew Zisserman. Bill Hoffman has been largely responsible for maintaining the software environment in which the work discussed in this paper has been implemented. CAR is funded by a Human Capital and Mobility grant from the European Community. JS was visiting the RobotVis project at INRIA from the Ecole Normale Supérieure de Lyon.

References

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:2–14, 1986.

[2] P. Beardsley, A. Zisserman, and D. Murray. Sequential update of projective and affine structure from motion. Technical Report OUEL 2012/94, Department of Engineering Science, University of Oxford, aug 1994.

[3] Andrea Califano and Rakesh Mohan. Systematic design of indexing strategies for object recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 709–710, New-York, NY, June 1993. IEEE Computer Society, IEEE.

[4] J. F. Canny. Finding edges and lines in images. Technical Report AI-TR-720, Massachusets Institute of Technology, Artificial Intelligence Laboratory, June 1983.

[5] Roberto Cipolla and Andrew Blake. The dynamic analysis of apparent contours. In *Proceedings of the Third International Conference on Computer Vision, Osaka, Japan*, pages 616–623, December 1990.

[6] D. T. Clemens and D. W. Jacobs. Model group indexing for recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 4–9, Maui, HI, June 1991.

[7] M. B. Clowes. On seeing things. *Artificial Intelligence Journal*, 2:79–116, 1971.

[8] R. Deriche, Z. Zhang, Q.-T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In J-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 of *Lecture Notes in Computer Science*, pages 567–576, Vol. 1, Stockholm, Sweden, May 1994. Springer Verlag.

[9] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., 1973.

[10] G.J. Ettinger. Large hierarchical object recognition using libraries of parameterized model sub-parts. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 32–41, Ann Arbor, MI, June 1988. Computer Society Press.

[11] David Forsyth, Joseph L. Mundy, Andrew Zisserman, Chris Coello, Aaron Heller, and Charles Rothwell. Invariant Descriptors for 3D Object Recognition and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):971–991, October 1991.

[12] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. In *FJCC*, volume 33, pages 291–304, 1968.

[13] C.G. Harris. Determination of Ego-motion from Matched Points. In *Proceedings of the 3rd Alvey Conference, Cambridge*, pages 189–192, September 1987.

[14] D. A. Huffman. Impossible objects as nonsense sentences. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 6, chapter ? Edinburgh University Press, 1971.

[15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, London, June 1987.

[16] Y. Lamdan and H.J. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In *Proceedings of the 2nd International Conference on Computer Vision, Tampa, Florida*, pages 238–249, December 1988.

[17] David Lowe. The Viewpoint Consistency Constraint. *The International Journal of Computer Vision*, 1(1):57–72, 1987.

[18] A. K. Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence Journal*, 4:99–118, 1973.

[19] R. Mohr, F. Veillon, and L. Quan. Relative 3d reconstruction using multiple uncalibrated images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 543–548. IEEE, 1993.

[20] Roger Mohr, Gunnar Sparr, and Olivier Faugeras. Multiple image invariants. In Hans Burkhart and Andrew Zisserman, editors, *ECCV92 Esprit BRA Workshop on Invariants for Recognition*, pages 109–132, Santa Margherita Ligure, Italy, May 1992.

[21] D.W. Murray, D.A. Castelow, and B.F. Buxton. From an image sequence to a recognized polyhedral object. *Image and Vision Computing*, 6(2):107–120, May 1988.

[22] L.G. Roberts. Machine perception of three-dimensional solids. In Tippett, J. and Berkowitz, D. and Clapp, L. and Koester, C. and Vanderburgh, A., editor, *Optical and Electrooptical Information processing*, pages 159–197. MIT Press, 1965.

[23] C. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen. Driving vision by topology. In *IEEE Computer Vision Symposium*, 1995. to appear.

[24] C. A. Rothwell, D. A. Forsyth, A. Zisserman, and J.L. Mundy. Extracting projective structure from single perspective views of 3D point sets. In *Proceedings of the 4th Proc. International Conference on Computer Vision*, pages 573–582, Berlin, Germany, May 1993. IEEE Computer Society Press.

[25] Charles Rothwell. Hierarchical object description using invariants. In Joseph L. Mundy, Andrew Zissermann, and David Forsyth, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 397–414, Ponta Delgada, Azores, October 1993. ESPRIT ARPA/NSF, Springer-Verlag.

[26] C.E. Springer. *Geometry and Analysis of Projective Spaces.* Freeman, 1964.

[27] F. Stein and G. Medioni. Structural Indexing: Efficient 3-D Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, February 1992.

[28] Kokichi Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, 1986.

[29] A. Sugimoto. Geometric invariant of noncoplanar lines in a single view. In *Proceedings of the International Conference on Pattern Recognition*, pages 190–195, Jerusalem, Israel, October 1994. Computer Society Press.

[30] D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings of the International Conference on Robotics and Automation, Raleigh, NC*, pages 208–220, 1987.

[31] D. Waltz. Understanding line drawings of scenes with shadows. In Patrick H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975.

[32] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow. 3D object recognition using invariance. Technical Report 2027/94, Oxford University Department of Engineering Science, November 1994. To appear in the AI Journal.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399