Understanding the Shape Properties of Trihedral Polyhedra

Charlie Rothwell and Julien Stern
INRIA, 2004, Route des Lucioles, Sophia Antipolis, 06902 CEDEX, France

Abstract. This paper presents a general framework for the computation of projective invariants of arbitrary degree of freedom (dof) trihedral polyhedra. We show that high dof. figures can be broken down into sets of connected four dof. polyhedra, for which known invariants exist. 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 which is based on the *butterfly invariant*. Furthermore, planar projective invariants can be measured which link together the subparts, and so we can develop a local-global description for general trihedral polyhedra. We demonstrate the recovery of polyhedral shape descriptions from images by exploiting the local-global nature of the invariants.

1 Introduction

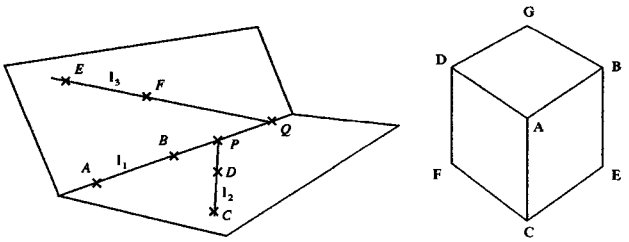
In this article we introduce a general scheme for understanding the shape properties of trihedral polyhedra. Trihedral polyhedra are solid polyhedra made up of planes in arbitrary positions, and as such, no special constraints exist between the planes. The nomenclature trihedral derives from the fact 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 would provide excess constraint and hence would not be generic (and stably realisable). The results in this paper are a summary of those given in [9].

In all, we generalise the result in [8] which showed how a projectively invariant description can be computed for four degree of freedom (dof) polyhedra from a single view. In turn, [8] was a extension of the work of Sugihara [11]. The latter dealt with scaled orthographic projection and the calibrated perspective cases, whereas the former demonstrated the projective equivalence of all members of the family of four dof. polyhedra generating a set of scene measurements using an uncalibrated camera. We show in this paper that the approach of [8] can be extended to include all trihedral polyhedra.

We also build on some recent work for computing the invariants of minimal point configurations in three-dimensional space. Being able to compute measures for small local feature groups provides robustness to occlusion. More global descriptions can be built up using the local-global nature of many shape descriptions [3, 10]. We derive a part-whole decomposition by drawing the invariant description of [8], and the invariants based on the *butterfly configuration* of Mundy [13] together. The butterfly invariant is a geometric description of a special six-point configuration.

Our interest in the butterfly invariant was promoted by the recent paper of Sugimoto [12]. This paper discusses an invariant very similar to the original butterfly invariant, but suggests an algebraic rather than a geometric formulation. However, Sugimoto suggested that the invariants in [12] in some way replace the invariants described by [8]. In fact, these two types of invariant can be taken hand-in-hand and are exactly complementary. This we show partly in this paper, and in more detail in [9].

The contributions of this paper are three-fold: in Section 2 we discuss how the original invariant description of [8] can be decomposed into a set of three independent butterfly invariants. Then we show in Section 3 how to reduce a five dof. figure into sets of



(a) (b)

Fig. 1. *(a) the butterfly consists of two sets of four coplanar points yielding a unique cross ratio. (b) the labelling of a cube used for the butterfly invariant computation.*

figures with four dof. For the simplest case of such an object (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 related through pairwise planar constraints. Finally, in Section 4 we report on algorithms for the extraction of the polyhedral descriptions and their image invariants.

2 The butterfly invariant

The butterfly is the simplest known invariant for a set of points in 3D space.[1] The configuration is composed of six points in space broken up into two four point groups, $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ and $\{\mathbf{A},\mathbf{B},\mathbf{E},\mathbf{F}\}$. Each four point group is coplanar, and two points are shared between the groups. This is shown in Fig. 1a. The invariant for the butterfly is measured through the construction of a cross ratio. As can be seen in Fig. 1a, it is possible to form a set of four collinear points and hence the invariant cross ratio [2]. These are the points $\{\mathbf{A},\mathbf{B},\mathbf{P},\mathbf{Q}\}$ from which we define the cross ratio $\tau = \{\mathbf{A},\mathbf{B};\mathbf{P},\mathbf{Q}\}$. In fact, we show in [9] that the same butterfly invariant actually takes the algebraic form:

$$\tau = \frac{|\mathtt{M}_{ACD}|}{|\mathtt{M}_{AEF}|}\cdot\frac{|\mathtt{M}_{BEF}|}{|\mathtt{M}_{BCD}|}, \tag{1}$$

which allows the direct computation of the invariant values from image data. Here $\mathtt{M}_{ijk}$ is the 3×3 matrix whose columns are the points i, j and k and $|\mathtt{M}|$ is the determinant of M. Although this invariant is very similar to that derived by Sugimoto [12] the interested reader will note that it can be derived more simply (shown in [9]). It is also worth noting that the form above was derived by Carlsson [1] using a more difficult, but in fact more general (and so elegant) approach based on the double algebra.

2.1 Computing polyhedral invariants with the butterfly

The projective invariants for polyhedral figures can be computed using the butterfly. Here we consider how to compute the invariants for a cube-like structure such as that shown in Fig. 1b. In fact, we see later that understanding the invariants for this type of figure is as far as we need to go to comprehend the invariants for all trihedral polyhedra. For a cube-like polyhedron, there are three independent invariants, for instance:

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

[1] That is for a configuration which satisfies no single geometric constraint.

It is simple to demonstrate that these three invariants are independent and form a basis for all of the other invariants. Thus a description based on the butterfly is equivalent to one which uses the invariants of [8]. Proofs of these statements are given in [9].

3 Simplifying higher order polyhedra

We now extend previous theories on trihedral polyhedra to the general case. This result is significant because it resolves a number of important but unanswered questions which were posed by [8]. 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 as shapes equivalent to cubes but with simple volumes cut-out of them. Note that the number of dof. of a polyhedra represent the dimensionality of the space of reconstructions which arise out of the image constraints.

In the following paragraphs we demonstrate that figures with more than four dof. can be decomposed into sets of related polyhedra which have only four dof. The first example we consider is the decomposition of five dof. figures into sets of four dof. figures. More complicated figures can be decomposed similarly. The simplification process means that the invariants 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 for general polyhedral structures are based on those for four dof. figures. However, no projective invariants of the complete structures exist as each four dof. subpart maintains a certain amount of (non-projective) independence from the other parts. We discuss this claim in Section 3.3.

Additionally, we can compute joint-invariants between the pairs of adjacent four dof. figures in 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, the subparts 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.

These two sets of invariants are all those which may be computed for an arbitrary trihedral polyhedron. 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 [8], 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 that they are made up of trihedral vertices and thus satisfy genericity conditions. Additionally, they are bounded solids which, due to self-occlusion under imaging, can be considered *not* to have interior volumes or surfaces. Thus, any *complete set* of adjacent planes forms a solid polyhedron. A complete polyhedron has three planes meeting at every vertex, and two planes at every edge. We can always render a set of adjacent planes complete using the trihedral constraint by defining additional planes passing through the edges and vertices.

We use a 3D coordinate system (X, Y, Z) and so planes satisfy: $a_j X_i + b_j Y_i + c_j Z_i + 1 = 0, j \in \{1, ..., n\}$ 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: $x_i = X_i/Z_i$ and $y_i = Y_i/Z_i$. Substituting these into the plane constraint equation, dividing through by Z_i and setting $t_i = 1/Z_i$ yields:

$$a_j x_i + b_j y_i + c_j + t_i = 0. \tag{3}$$

Here we have assumed a camera calibration; as stated in [8], this need not be the case. The trihedral assumption means that each point lies on three planes, say $j \in \{\alpha, \beta, \gamma\}$. Eliminating the t_i between pairs of constraints yields two linear equations per point:

$$\begin{aligned}(a_\beta - a_\alpha)x_p + (b_\beta - b_\alpha)y_p + (c_\beta - c_\alpha) = 0,\\ (a_\gamma - a_\alpha)x_p + (b_\gamma - b_\alpha)y_p + (c_\gamma - c_\alpha) = 0.\end{aligned} \tag{4}$$

Now, from the m image vertices observed in each image, we form $\mathtt{A}(\mathbf{x}, \mathbf{y})\ \mathbf{w} = 0$ which is a matrix equation of $2m$ constraints in $3n$ unknowns. Here $\mathbf{x} = (x_1, \ldots, x_m)^\top$, $\mathbf{y} = (y_1, \ldots, y_m)^\top$, $\mathtt{A}(\mathbf{x}, \mathbf{y})$ is a $2m \times 3n$ matrix, and $\mathbf{w} = (a_1, b_1, c_1, a_2, \ldots, c_n)^\top$. We 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 polyhedron imaged by a single pinhole camera.*

Proof See [8] and more completely [7]. □

The kernel of A can be represented as $\mathbf{w} = \sum_{i=1}^{d} \lambda_i \mathbf{b}_i$. We can orient the basis for $\mathbf{w}$ so that $\mathbf{b}_i = (\delta_1, \delta_2, \delta_3, \ldots, \delta_1, \delta_2, \delta_3)^\top$ for $1 \leq i \leq 3$ and where $\delta_i = 1$, and $\delta_{mod_3(i+1)} = \delta_{mod_3(i+2)} = 0$. We may also fix the other basis vectors so that:

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

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 of the reconstructed polyhedron has coordinates $(\lambda_1, \lambda_2, \lambda_3, 1)^\top$.

3.2 Four dof. polyhedron

It was shown in [7] 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.

3.3 Five dof. polyhedron

Consider the case for $d = 5$. To simplify the discussion we represent $\mathbf{b}_4$ and $\mathbf{b}_5$ by:

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

Separating out the first four dof. figure

We return to the form of the constraints given in eqn (4). 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. Consider writing $\alpha = (\lambda_1, \lambda_2, \lambda_3)^\top$. Then, making use of the fact that both $\mathbf{r}_1$ and $\mathbf{s}_1$ 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 (4) reduces to the following vectorial constraints:

$$(x_p\ y_p\ 1)^\top . (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i) = 0, \quad \text{and} \quad (x_q\ y_q\ 1)^\top . (\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i) = 0. \tag{5}$$

These define $(\lambda_4 \mathbf{r}_i + \lambda_5 \mathbf{s}_i)$ uniquely up to a scale as $\mathbf{x}_p$ and $\mathbf{x}_q$ are distinct points. This solution exists for all λ_4 and λ_5, and so either $\mathbf{r}_i$ is parallel to $\mathbf{s}_i$, or one of $\mathbf{r}_i$ and $\mathbf{s}_i$ is null (but not both). Now, within the bounds of the geometric constraints used so far, we

are free to orient $\mathbf{b}_4$ and $\mathbf{b}_5$ as we desire. In order to simplify the overall description, we make a further change of basis. Assuming 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 change of basis (perhaps trivially) to ensure that $\mathbf{r}_i$ is non-zero and $\mathbf{s}_i$ is null. Once done, neither the first plane, nor plane i, will depend on λ_5. By reapplying the reasoning, we may extend the process and provide independence to λ_5 for all of the other planes which 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 depends only on λ_4. Thus the new plane depends only on λ_4 (and so its 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 which consequently represent a four dof. figure.

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 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:

1. The planes in $\mathcal{S}_1$ use up four of the five dof. for the polyhedron. This is because the planes in $\mathcal{S}_1$ are defined by the 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 in $\mathcal{S}_1$ which is adjacent to $\mathcal{S}$. To do this, we again change basis so that this 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 on only 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 chosen so that is shares an edge with the first plane. Thus we have:

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

We now repeat the process used previously for the elimination of the components of $\mathbf{s}_i$ 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 (5)). We may now 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:

$$\begin{aligned}\mathbf{b}_4 &= (0,0,0,\mathbf{r}_2^\top,\ldots,\mathbf{r}_{m-1}^\top,0,0,0,\ldots,0,0,0,\mathbf{r}_k^\top,\ldots,\mathbf{r}_n^\top)^\top,\\ \mathbf{b}_5 &= (0,0,0,\ldots\ldots,0,0,0,\mathbf{s}_m^\top,\ldots\ldots,\mathbf{s}_{k-1}^\top,\mathbf{s}_k^\top,\ldots,\mathbf{s}_n^\top)^\top.\end{aligned} \tag{6}$$

Such a basis represents a decomposition of the polyhedron into two four dof. complete solid polyhedra built around the first plane (the first polyhedron of planes 1 to $(m-1)$, and the second of plane 1, and planes m to $(k-1)$). The first plane is called the *common* plane. In addition to these sub-figures, there may be a set of planes k to n which we represent by the set $\mathcal{S}_3$. $\mathcal{S}_3$ is empty for simple polyhedra, and so $k = n+1$. However, this is not the general case and $\mathcal{S}_3$ can well contain further polyhedral subparts. Note that the planes in this subsidiary set are still constrained completely by λ_4 and λ_5.

$\mathcal{S}_3$ at first has the appearance of being a more complex figure than the four dof. subparts which we have already extracted, though this is not really the case. $\mathcal{S}_3$ will always be composed of other four dof. polyhedra. The simplest way to understand this is to reorder the planes so that one of the planes in $\mathcal{S}_1$ adjacent to $\mathcal{S}_3$ is set to the first plane of w (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 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 thus extend our current understanding of trihedral polyhedra via:

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, and more completely in [9]. □

Projective inequivalence of five dof. figures

Unlike the four dof. figures discussed previously in [8], two different reconstructions of a five dof. figure need not necessarily be projectively equivalent. Formally:

Theorem 3.3 *The different reconstructions of a five dof. non-trivial polyhedron imaged by an uncalibrated pinhole camera are not in the same projective equivalence class.*

Proof Given in [9]. □

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 [8]. As stated in Section 2.1, these can be measured using the butterfly invariants. We can also measure a number of invariants in the common plane from the coplanar edges of each adjacent sub-polyhedra. Generically the number of invariants of this form depends on the 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 $\sum_{i=1}^{j}(2m_i - 8)$ computable planar invariants. It can be shown that the butterfly and planar invariants represent all of the invariants which can be computed for such a figure.

3.5 Higher degree of freedom figures

We can use the results of Section 3.3 to complete the description paradigm for polyhedra of 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. The computation of the invariants for the entire object is then straightforward:

- Compute projective invariants for the planes within each subpart.
- Compute planar invariants between the edges in all the common planes.

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

- Use an edgel detector to initialize a polyhedral snake on sets of image features. Then compute invariants for the snake.
- Search for pairs of adjacent closed regions in the image suitable for estimating 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 edgel and line extraction. The edgel detector we use is described in [6] and ensures good connectivity around image junctions. Straight lines are fitted to edgel-chains using orthogonal regression and a topological (connectivity) structure composed of connected straight lines and edgel-chain segments is produced. The actual data structures we use are built around a *vertex-edge-face* topology hierarchy [5]. Vertices are typically used to represent junctions and the interfaces between pairs of lines or edgel-chains. Edges link vertices and are instantiated geometrically by lines and edgel-chains. Faces represent 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 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.

The first phase of processing involves the extraction of edges, as demonstrated in Fig. 2. Notice that we have been able to recover a fairly good description of the polyhedron's topology with the edge detector. Principally the edge description consists of three closed regions each of which match projected polyhedral faces. Being able to find these image regions means that we can reject scene clutter (non-polyhedral regions) rapidly

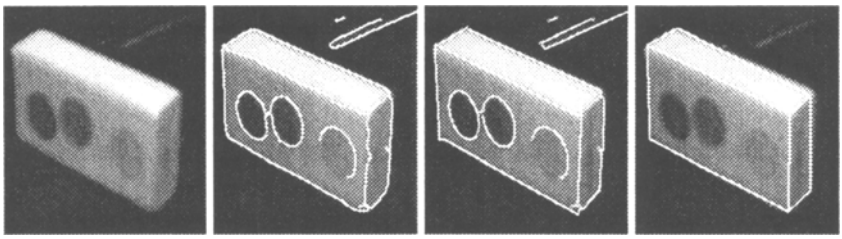


Fig. 2. *First we show a polyhedron and then the output of the edgel detector superimposed. Note that the basic topological structure of the edges is correct. Next are the lines fitted by orthogonal regression and finally the polyhedral snake.*

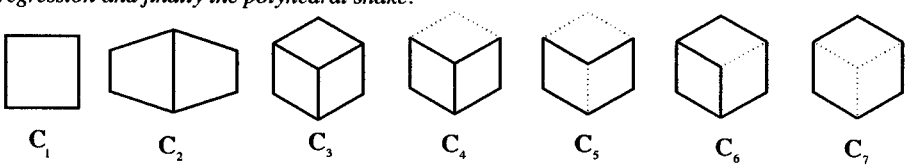


Fig. 3. *The topological descriptions stored for the projections of a six-plane polyhedron.*

during subsequent processing. We then fit lines and throughout ensure the maintenance of the topological description. After fitting, we extract faces from the image. For Fig. 2 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 faces and on the lines which they contain. However, over-segmentation can arise due to the edge detector recovering small noisy features. A number of these are shown in Fig. 2. 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 face 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 face 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. Consecutive line segment endpoints should also be reasonably close as polyhedral faces are generally polygonal. Given this, we can discard the edgel-chains between lines and reduce the face representations to ideal polygons.

The penultimate stage of processing involves the matching of the adjacency graph of the faces to topological models which we have for the polyhedra in the model base. The topological descriptions of the lines and the snake are consistent in Fig. 2. The complete set of descriptions from which we form snakes is given in Fig. 3. 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 [4]. The final position of the snake yields sufficient structure for invariant computation. In Fig. 2, all of the butterfly invariants are unity because the polyhedron is projectively equivalent to a cube (which is composed of three of sets of parallel planes). Table 1 shows how the three invariants for the shape change over the iterations of the snake.

A similar example of polyhedron extraction using snakes is in Fig. 4. In Fig. 5 we have a harder case. The edgel detector failed to recover the polyhedron's 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 C_6 (in Fig. 3) and so we are able to hypothesize the location of the missing edge by joining up a pair of vertices. We return to the complete snake description (C_3) once the missing edge has been inserted. In

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. 2 for the first eight iterations. The invariant values should be all converge to unity.*

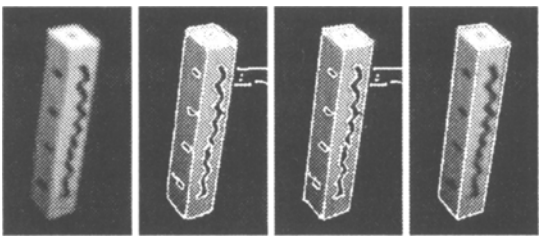


Fig. 4. *The polyhedron, superimposed edgels, fitted lines, and polyhedral snake.*

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. One finding a lone quadrilateral, such as C_1, we terminate processing.

4.2 Finding polyhedra using local butterfly invariants

The above approach is global; we now describe a local method which uses the butterfly invariant. The initial processing is the same in that we extract edgels and lines, and then recover *faces* from the image which represent closed regions. 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 paired to produce butterflies. 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 [10]. Briefly, 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 are the four dof. descriptions of the polyhedra in the images. In

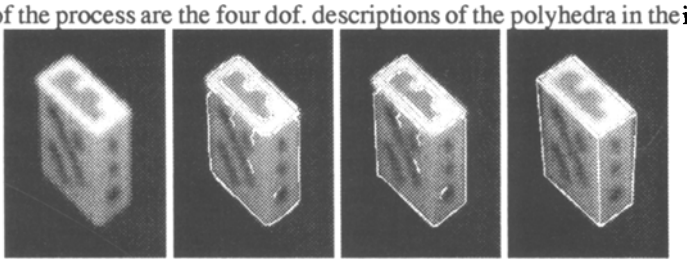


Fig. 5. *First the polyhedron and the edgels. The complete topological structure of the polyhedron has not been recovered. Next are the lines and finally the polyhedral snake.*

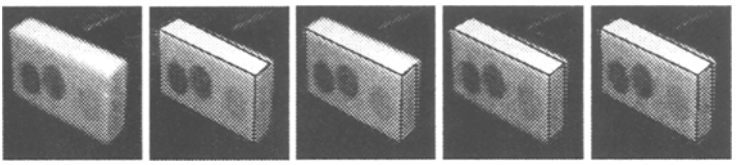


Fig. 6. *After segmentation and closed region detection we extract three butterflies with invariants approximately equal to unity. These butterflies can be merged into a single polyhedral description.*

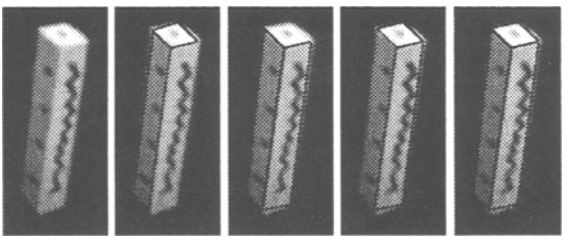


Fig. 7. *We recover three butterflies for this imaged polyhedron.*

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.

Extraction of the butterflies is shown in Fig. 6. We show the three different extracted butterflies for the image section in the figure. The butterflies are composed 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 match butterfly invariants for objects which have projective equivalence to a cube. They can thus be merged into the single polyhedral description shown on the right which is a single four dof. polyhedron.

Another example is in Fig. 7. There the three butterflies have invariants equal to 1.000, 1.000, and 1.029. However, in Fig. 8 we show an example in which only a single butterfly can be recovered (with an invariant of 1.000) due to the failure of the edgel 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.

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

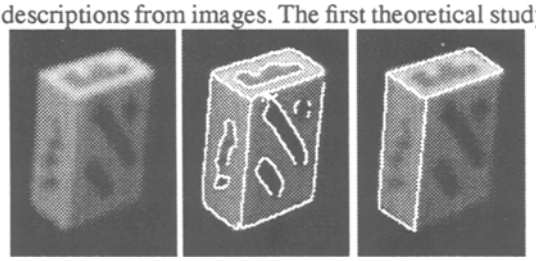


Fig. 8. *For this polyhedron we recover only two polyhedral faces as closed regions in the edgel description. Thus we can find only a single butterfly configuration. However, the invariant value for the butterfly matches precisely that which we would expect to find for such a polyhedron.*

simple algebraic formulation for the butterfly invariant and showed how it is related to the polyhedral invariants of [8].

The second theoretical contribution is a completion of the projective invariant description of [8] to trihedral polyhedra of arbitrary degrees of freedom. We demonstrated that all higher order polyhedra can be broken down into sets of four dof. polyhedra associated using plane projective invariants. This yields a complete description paradigm for arbitrary trihedral polyhedra.

Finally, we developed two different implementations of algorithms intended for the detection of polyhedra in images. 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.

A more complete version of this paper [9] is available as a technical report from the URL: <URL ftp://ftp.inria.fr/INRIA/publication/publi-ps-gz/RR/RR-2661.ps.gz>.

Acknowledgments

We acknowledge the input of a number of people in this work, principally: Olivier Faugeras, David Forsyth, Bill Hoffman, Joe Mundy, and Andrew Zisserman. Thanks to Pippa Hook for proof-reading this article. CR is funded by a Human Capital and Mobility grant from the European Community. JS was visiting INRIA from the Ecole Normale Supérieure de Lyon.

References

[1] S. Carlsson. Multiple image invariance using the double algebra. In *Applications of Invariance in Computer Vision*, volume 825 of *LNCS*, p.145–164. Springer-Verlag, 1994.

[2] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[3] G. Ettinger. Large hierarchical object recognition using libraries of parameterized model sub-parts. *Proc. CVPR*, p.32–41, 1988.

[4] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Proc. ICCV*, p.259–268, 1987.

[5] C. Rothwell, J. Mundy, and W. Hoffman. Representing objects using topology. In preparation, 1996.

[6] C. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen. Driving vision by topology. *Proc. IEEE International Symposium on Computer Vision*, p.395–400, 1995.

[7] C. Rothwell, D. Forsyth, A. Zisserman, and J. Mundy. Extracting projective information from single views of 3D point sets. TR 1927/92, Oxford Univ. Dept Eng. Sci., 1992.

[8] C. Rothwell, D. Forsyth, A. Zisserman, and J. Mundy. Extracting projective structure from single perspective views of 3D point sets. *Proc. ICCV*, p.573–582 1993.

[9] C. Rothwell and J. Stern. Understanding the shape properties of trihedral polyhedra. TR 2661, INRIA, 1995.

[10] C. Rothwell. *Object recognition through invariant indexing.* Oxford University Press, 1995.

[11] K. Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, 1986.

[12] A. Sugimoto. Geometric invariant of noncoplanar lines in a single view. *Proc. ICPR*, p.190–195, 1994.

[13] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow. 3D object recognition using invariance. TR 2027/94, Oxford Univ. Dept Eng. Sci., 1994, to appear the AI Journal.