Algebraic polyhedral constraints and 3D structure from motion

D W Murray

The application of algebraic polyhedral constraints to the computation of the 3D structure and motion of polyhedral objects is described. The method, which works when complete 2D line-drawing information is available, guarantees the recovery of planar faces. The normals to these faces are used for matching to models. Several examples are given to illustrate the scope of the method.

Keywords: algebraic polyhedral constraints, 3D structure, polyhedral objects

Murray et al. describe a motion processing system, ISOR, which is able to recover the 3D motion and structure of polyhedral objects from an image sequence and goes on, where possible, to recognize the object as one from a database of object models. The system performs a ‘bottom-up’ pass through a vision processing hierarchy in four stages: low level – compute visual motion at intensity edges in a sequence of time-varying imagery; segmentation – segment the edges (and thereby visual motion) into groups lying on the same straight edges in the image; structure-from-motion (SFM) – compute the 3D structure and motion of the partial wireframe of which the linked straight edges in the image are the perspective projection, and; recognition – match the 3D partial wireframe to a complete wireframe stored in a database of object models.

Here we make two modifications. First, the computation of 3D structure from 2D visual motion is made under algebraic constraints which force the reconstructed partial wireframe in 3D to be strictly polyhedral. The motivation is to make surfaces explicit at an earlier stage of the processing, in particular before model matching. One way of achieving this would be to fit planes through the 3D edges and vertices computed by the existing, unconstrained algorithm. If sets of edges were adjudged coplanar, it would then be possible to re-execute the algorithm with these additional constraints. However, such a method neglects the structural information contained in a single image of a polyhedral scene. This work utilizes the information from line-drawing analysis to provide a priori constraints to the SFM computation, using the techniques described by Sugihara. The second modification is that we match using the recovered planar surfaces as primitives, using the search method of Grimson and Lozano-Pérez and the geometrical constraints described by Murray.

UNMODIFIED SFM METHOD

For the present purpose, only details of the third stage of the system, where 3D scene structure is computed from 2D visual motion, are of direct concern. Prior to a résumé of that stage, brief details of the earlier stages are given to clarify what information is explicit (see Reference 1 for a fuller discussion).

Computing visual motion

The images in a sequence are processed in groups of three. The second of the group is regarded as the ‘current’ frame, and those captured one time interval earlier and one interval later are defined as the backward and forward frames, respectively. Intensity edges are computed in all three frames using the Canny detector, which provides the position \( r_e = (x_e, y_e)^T \) of each edge \( e \) in the image to subpixel precision, along with its orientation and strength (change in grey value). A thresholding operation filters out weak, isolated edges. Visual motion is computed at each edge in the central frame by analysing matching strength distributions between edges in consecutive frames. Sitting at an edge \( e \) in the current frame, a search is made around the position \( r_e \) in the forward frame for edges \( f \) to which to match, and initial matching strengths between \( e \) and \( f \) at \( r_e \) are defined using a similarity measure which favours matching between edges of similar strength and orientation. These initial strengths are improved using neighbourhood support within an iterative relaxation scheme.
Figure 1. (a) ‘Current’ image of a toy truck from a sequence where the camera translates towards the camera; (b) visual motion components

A similar search is made in the backward frame, and the probability distributions combined simply by time-reversing the backwards displacements. The resulting distribution around position $r_e$ is analysed using a principal axis decomposition, yielding two orthogonal vector components of visual motion and associated confidences. Figure 1a shows the ‘current’ image of a sequence of a toy truck approaching the camera, and the higher confidence components of visual motion are shown in Figure 1b.

Note that along the extended edges the aperture problem prevails, and the major components are mostly normal to the edge direction. In this situation the lower confidence minor tangential components are of such little statistical worth that they may be discarded.

**Segmentation**

This stage segments the visual motion into groups lying along extended straight edges. Because the visual motion is at edges, it suffices (in simple worlds!) to segment the edge positions, with no reference to the visual motion per se. The segmentation proceeds through edgel linking into extended strings, breaking the strings into sections at points of high curvature, and by determining which sections comprise straight edges. Attempts are then made to link up straight edges which appear to converge to a single vertex. Figure 2 shows the final result for the truck, where the circles indicate vertices. As the visual motion is computed at the edgels, it is a trivial matter to import the visual motion into the segmentation graph.

**Unmodified SFM algorithm**

The unmodified SFM algorithm is founded on the assumptions that, first, each subgraph is the projection of a rigidly moving object in the scene and, second, that straight edges and vertices in the image map to straight edges and vertices in the scene.

The scene can therefore be (over-) described by $n + 6$ parameters $(\xi_1, \ldots, \xi_n, \psi, \Omega)$ where $\xi_i$ is the inverse or reciprocal depth of the scene vertex which projects to image endpoint $i$, $\psi$ is the translational velocity relative to the camera, and $\Omega$ is the instantaneous angular velocity relative to the camera. These scene parameters are varied so as to minimize:

$$D = \sum_e w_e (|v_e| - |v_e^{\text{pred}}|)^2$$

where $v_e$ is the measured (major) component of visual motion at edgel $e$, $v_e^{\text{pred}}$ is the predicted component, and $w_e$ is the confidence associated with the measurement. The remainder of this section explains how $v_e^{\text{pred}}$ is derived in terms of the unknown scene parameters and known image quantities.

The over-determination arises as, without external knowledge, it is impossible to derive more than $n + 5$ of the parameters because of the inevitable depth/speed scaling ambiguity in monocular motion processing. There are two obvious ways of reducing the dimensionality of the parameterization: by fixing one of the reciprocal depth values, or by fixing the magnitude of the translational velocity.

Figure 3 sketches the scene and camera geometries under consideration. Consider the image endpoint $i$ at $r_i$. It is related to the corresponding scene point $R_i$ by $r_i = -l \xi_i R_i$, where $l$ is the focal length of the camera, and $\xi_i = 1/(R_i, z)$. The full projected motion at $r_i$ is the time differential:

$$\dot{r}_i = -l \xi_i (R_i - R, \xi_i R_i, \dot{z})$$

Figure 2. Segmentation for the truck

Figure 3. Camera and scene geometries
The motion of the scene point can always be expressed as:
\[ \mathbf{\ddot{r}}_s = \mathbf{v} + \Omega \times \mathbf{r}_s \]
so that after substitution:
\[ \mathbf{\ddot{r}}_s = -l \mathbf{\xi}_s (\mathbf{v} \cdot \mathbf{r}_s) \mathbf{r}_s + \Omega \times \mathbf{r}_s \]
\[ \mathbf{\ddot{r}}_s = -l \mathbf{\xi}_s (\mathbf{v} \cdot \mathbf{r}_s) \mathbf{r}_s + \frac{(\Omega \times \mathbf{r}_s) \cdot \mathbf{\ddot{r}}_s}{l} \]
Now consider a point \( \mathbf{r} \) on the straight edge between endpoints \( \mathbf{r}_1 \) and \( \mathbf{r}_2 \):
\[ \mathbf{r} = \lambda \mathbf{r}_1 + (1-\lambda) \mathbf{r}_2 \quad 0 \leq \lambda \leq 1 \]
The visual motion at this point must be just:
\[ \mathbf{\ddot{r}} = \lambda \mathbf{\ddot{r}}_1 + (1-\lambda) \mathbf{\ddot{r}}_2 \]
This is almost the data we measure and wish to predict, but there are two details which must be taken care of. First, the edgel position \( \mathbf{r}_i \) will probably not lie directly on the line between \( i \) and \( j \); edgels will meander either side of the true line. To overcome this, we estimate \( \lambda \) to be that describing the nearest point on the straight line, given an edgel at \( \mathbf{r}_s \) between endpoints \( i \) and \( j \):
\[ \lambda_s = \frac{(\mathbf{r}_i - \mathbf{r}_s) \cdot (\mathbf{r}_j - \mathbf{r}_s)}{|(\mathbf{r}_j - \mathbf{r}_i)|^2} \]
and hence the predicted full visual motion at the edgel is:
\[ \mathbf{\dot{v}}^{pred}_s = \mathbf{\dot{r}}_s + (1-\lambda_s) \mathbf{\ddot{r}}_s \]
Second, we wish to derive a component of \( \mathbf{\dot{v}}^{pred}_s \). This is found simply by vector projection onto the measured component \( \mathbf{v}_m \), i.e. our predicted value of the component is:
\[ \mathbf{v}^{pred}_s = \mathbf{v}_m (\mathbf{\dot{v}}^{pred}_s, \mathbf{v}_m) / |\mathbf{v}_m|^2 \]
After some routine working, the magnitude of the predicted component is given by:\
\[ |\mathbf{v}^{pred}_m| = |\mathbf{v}_m| \left[ \mathbf{\xi}_s (\mathbf{\dot{r}}_s - (\mathbf{\ddot{r}}_m + \mathbf{\dot{r}}_s)) \cdot \cos \theta + \mathbf{\xi}_s (\mathbf{\dot{r}}_s - (\mathbf{\ddot{r}}_m + \mathbf{\dot{r}}_s)) \cdot \sin \theta + \mathbf{\xi}_s (\mathbf{\dot{r}}_s - (\mathbf{\ddot{r}}_m + \mathbf{\dot{r}}_s)) \cdot \mathbf{\ddot{r}}_m + \Omega \cdot \mathbf{\xi}_s (\mathbf{\dot{r}}_s - (\mathbf{\ddot{r}}_m + \mathbf{\dot{r}}_s)) \cdot \mathbf{\ddot{r}}_m \right] \]
where \( \cos \theta = (\mathbf{v}_m, \mathbf{\dot{v}}^{pred}_s) / |\mathbf{v}_m| \), \( \sin \theta = (\mathbf{v}_m, \mathbf{\ddot{v}}^{pred}_s) / |\mathbf{v}_m| \), and where \( \mathbf{f}_j = x_j \cos \theta + y_j \sin \theta \), \( g_j = y_j \cos \theta - x_j \sin \theta \), and similarly for \( f_j \) and \( g_j \).

**POLYHEDRAL CONSTRAINTS**

Although there is an implicit polyhedral assumption in the existing SFM algorithm, in that we consider a 3D scene to be made up of straight edges linked by vertices, nowhere do we exploit the fact that the straight edges lying around a face should be coplanar. To impose this, though, obviously requires that we discover which edges comprise the border of the face. This can be achieved by analysing the 2D line drawing, at least providing it is complete, a process which also provides other clues about relative depth. The two major methods of reconstructing polyhedra from 2D line drawings are due to Kanade\(^6\), who recovered shape from line drawings using a gradient space approach, and Sugihara\(^2,3\), who developed linear algebraic constraints imposed in real space. Sugihara’s technique has advantages over that of Kanade. First, the former’s constraints impose necessary and sufficient conditions that the object is a polyhedron, whereas the latter’s apply only a necessary condition. Second, gradient space techniques appear more sensitive to errors in 2D vertex positions than the algebraic constraints. Here we utilize Sugihara’s method but, unlike previous published experimental work, we apply the constraints under perspective projection.

**Sugihara’s algebraic constraints**

Using the information in the 2D graph derived for segmentation, we first create a labelled 2D line drawing with lines corresponding to convex edges labelled ‘+’, those corresponding to concave edges labelled ‘−’, and those corresponding to occluding edges labelled ‘>’, where the arrow points that the area to the right of the arrow is the occluding face.

Following Sugihara\(^2,3\), let \( \mathcal{P} \) be the set of visible vertices, so that \(|\mathcal{P}| = n\), and let \( \mathcal{F} \) be the set of (partially or wholly) visible faces, with \(|\mathcal{F}| = m\). Now define the set \( \mathcal{R} \) as \( \mathcal{R} \equiv \mathcal{F} \times \mathcal{P} \) for \( (v,f) \in \mathcal{R} \) if \( v \in \mathcal{P} \), and each \((v,f) \in \mathcal{R}\) is called an incidence pair and the triple \( (v,f) \in \mathcal{R} \) is an incidence structure. This is easily computed from the labelled line drawing.

Define scene points \( \mathbf{R} \) lying on the face \( f \) by:
\[ \mathbf{R} \cdot \mathbf{N}_f = -1 \]
where \( \mathbf{N}_f \) is normal to the face and sticks out of the surface into free space. Using the perspective projection (equation (3)) and writing \( \mathbf{N}_f = (a_i b_j c) \), each \((v_i, f_i) \in \mathcal{R}_f \) gives rise to an equation:
\[ -a_i x_i / l - b_i y_i / l + c_i + \xi_i = 0 \]
Collecting these together for every incidence pair in \( \mathcal{R}_f \) results in the system:
\[ \mathbf{A} \mathbf{s} = \mathbf{0} \]
where:
\[ \mathbf{s} = (a_1 b_1 c_1 \ldots a_m b_m c_m \xi_1 \ldots \xi_n)^T \]
is an unknown column vector of length \((3m + n)\) and \( \mathbf{A} \) is a known \(|\mathcal{R}_f| \times (3m + n)\) matrix.

Any face \( f_i \) divides space into two. If a point \( \mathbf{R}' \) is such that \( \mathbf{R}' \cdot \mathbf{N}_f + 1 > 0 \) then the point lies in front of the plane of the face, and if \( \mathbf{R}' \cdot \mathbf{N}_f + 1 < 0 \) it lies behind the plane. Now consider two faces \( f_i \) and \( f_j \) sharing a concave edge, as illustrated in Figure 4a. Consider the vertex \( v_i \) such that \((v_i, f_i) \in \mathcal{R}_f \) but \((v_i, f_j) \notin \mathcal{R}_f \)

Clearly:
\[ -a_i x_i / l - b_i y_i / l + c_i + \xi_i > 0 \]
But suppose these faces share a convex edge (see Figure 4b). Then:

\[ +a_{x_i}/l + b_{y_j}/l - c_j + \xi_i > 0 \]  (16)

In fact, the situation is a little more complicated. The analysis above is only straightforwardly applicable when the joining edge is not a re-entrant edge on a non-convex face. In practice a test is made (in 2D) to see whether all the vertices of at least one face lie to one side of the line created by extending the shared edge. If they do, then the correct inequality can be chosen. For example, in Figure 4c all the vertices of face \( f \) lie on one side, so we can easily decide that vertices \( v_5 \) and \( v_b \) lie in front of, and behind, the plane of \( f \), respectively. Consider Figure 4d, where \( f_k \) occludes \( f \). Let \( v_m \), \( v_q \) and \( v_p \) be the initial, end and midpoint points of the occluding edge, following along the label direction. (Note that \( v_5 \) is not an obvious member of \( \mathcal{F} \). Sugihara explains that such pseudo-vertices are added to \( \mathcal{F} \) and the pseudo-incidence pair \( (v_5, f_k) \) to \( \mathcal{R} \) during creation of the incidence structure. They are then treated just like any other members.)

Either none or one of \( v_m \) or \( v_p \) could touch \( f \), but not both (if both touch, the line must be labelled concave, not occluding). Thus, three constraints become available:

\[-a_{x_i}/l - b_{y_j}/l + c_j + \xi_p \geq 0 \]  (17)
\[-a_{x_i}/l - b_{y_j}/l + c_j + \xi_\sigma \geq 0 \]  (18)
\[-a_{x_i}/l - b_{y_j}/l + c_j + \xi_p \geq 0 \]  (19)

Unfortunately, we cannot apply these constraints within a single SFM computation, because of the possibility that the occluding and occluded objects move differently, although they can of course be used to constrain depths between separate applications of the SFM algorithm.

Constraints of the type shown in equations (17) and (18) (and indeed the occlusion constraints, if used) can be expressed as:

\[ B_s > 0 \]  (20)

(where, for occlusions, the inequality sometimes permits equality).

Hence, given that we wish to recover a polyhedral object, one might post the structure from motion calculation as:

\[ D = \sum R(v) \left( \| v - \| v^{opt} \| \right)^2 \]  (21)

subject to the conditions

\[ A_s = 0 \]  (22)
\[ B_s > 0 \]  (23)

However, Sugihara highlights several difficulties with applying the constraints naively. The principal one is that not all the equations in the equality constraint (equation (22)) are linearly independent. The steps used by Sugihara\(^2,3\) to eliminate this problem are outlined below.

Eliminating dependent constraints

Because only a few of the vertex positions in a polyhedron are independent, some of the constraints expressed by the set \( \mathcal{R} \) depend on others. It is necessary both to eliminate these dependent constraints and to elicit the set of independent vertices\(^2,3\).

First, it is necessary for the image vertices \( (x_1, y_1, \ldots, x_n, y_n) \) to be in general position, i.e. they must be algebraically independent over the rational field so that there are no special relationships between their positions (e.g. three vertices must not always be collinear, nor three edges concurrent). Given this condition, we seek a position-free incidence structure \( S \), one where the constraint system has a non-trivial solution when the vertices are in general position. Sugihara proves the following:

**Theorem 1.** If \( S = (\mathcal{F}, \mathcal{R}, \mathcal{S}) \) is an incidence structure in which no three faces sharing a vertex have a common line of intersection, then \( S \) is position-free if and only if for all \( \mathcal{X} \subseteq \mathcal{F} \) and \( \mathcal{X} \neq \emptyset \):

\[ \left| \mathcal{F}(\mathcal{X}) \right| + 3 \left| \mathcal{X} \right| \geq |\mathcal{R}(\mathcal{X})| + 4; \]

where \( \mathcal{F}(\mathcal{X}) \) is the set of vertices that are on some faces in \( \mathcal{X} \) and \( \mathcal{R}(\mathcal{X}) \) is the set of incidence pairs involving elements of \( \mathcal{X} \).

**Theorem 2.** If \( S \) as described in Theorem 1 is position-free and the vertices are in general position, then the system \( A_s = 0 \) is linearly independent.

Given some set of incidence pairs \( \mathcal{R} \), we can use theorem 1 to test whether it is position free. If it is not, we search for a maximal set \( \mathcal{R}^* \subseteq \mathcal{R} \) for which the reduced incidence structure \( S^* \) is position-free by testing that for all \( \mathcal{X} \subseteq \mathcal{F} : \left| \mathcal{X} \right| \geq 2:

\[ \left| \mathcal{F}^*(\mathcal{X}) \right| + 3 \left| \mathcal{X} \right| \geq |\mathcal{R}^*(\mathcal{X})| + 4 \]

where \( \mathcal{R}^*(\mathcal{X}) \) is the subset of \( \mathcal{R}^* \) involving elements of \( \mathcal{X} \) and \( \mathcal{F}^*(\mathcal{X}) = \{ v \in \mathcal{F} : \{ v \} \times \mathcal{X} \} \cup \mathcal{R}^* \neq \emptyset \).

Let the reduced matrix associated with the constraints in \( \mathcal{R}^* \) be \( A^* \). Theorem 2 indicates that it must
be possible to transform $A'$ by appropriate column permutation into $A'$, which may be partitioned so that:

$$A's' = (A_1|A_2)s' = 0$$  \hspace{1cm} (25)$$

where $A_1$ is a non-singular $|\mathcal{A}^*| \times |\mathcal{A}^*|$ matrix whose inverse therefore exists. The vector $s'$ has the same members as $s$, but certain of the $x$ values will have been permuted. Splitting $s'$ into two vectors $s'=(\eta, \xi)'$, it is possible to write:

$$\eta = -A_1^{-1}A_2\xi$$  \hspace{1cm} (26)$$

It is clear that we may associate the vector $\xi$ with the reciprocal depths of the independent vertices, and $\eta$ with the other, dependent reciprocal depths and plane parameters. The number of independent parameters is $|\xi| = 3m + n - \text{rank}(A_1)$.

**Finding the independent set of vertices**

It was shown above that a set of independent vertices must exist. Here we briefly indicate the method proposed by Sugahara to find such a set, and thus show how to find the permutation of columns that transforms $A'$ to $(A_1|A_2)$, $s$ to $s'$, and $B$ and $B'$ (used later).

It is possible to define the degree of freedom $\sigma_d(\mathcal{Y})$ of a set of vertices $\mathcal{Y} \subseteq \mathcal{T}$ such that the pair $(\mathcal{T}, \sigma_0)$ is a matroid. The subset of vertices we require is that which is the maximal independent subset of $\mathcal{T}$, that is a base of the matroid. Sugahara proves the following:

**Theorem 3.** If $S^* = (\mathcal{T}; \mathcal{F}; \mathcal{R}^*)$ is a position free incidence structure then $\mathcal{Y} \subseteq \mathcal{T} - \mathcal{T} - \mathcal{R} - \mathcal{R}^*$ is an independent set of the matroid $(\mathcal{T}; \sigma_0)$ if and only if for all $\mathcal{F} \subseteq \mathcal{T}$:

$$|\mathcal{T}^*(\mathcal{F})| = 3.\vert \mathcal{F}\vert \geq |\mathcal{R}^*(\mathcal{F})| + |\mathcal{T}^*(\mathcal{F}) \cap \mathcal{Y}|$$

Using this, and the fact for any $\mathcal{Y} \subseteq \mathcal{T}$, $\sigma_d(\mathcal{Y}) = \max(\{ |\mathcal{Y}'| \})$ such that $\mathcal{Y}' \subseteq \mathcal{Y}$ and $\mathcal{Y}'$ is an independent set of matroid $(\mathcal{T}; \sigma_0)$, we can build an independent subset $\mathcal{Y}$ by choosing vertices $v$ one-by-one from $\mathcal{T} - \mathcal{T} - \mathcal{R} - \mathcal{R}^*$. Starting with $\mathcal{Y} = \emptyset$ we test whether $v$ is independent using theorem 1. If it is, $\mathcal{Y} \rightarrow \{v\}$ is independent, otherwise $v$ is discarded. As soon as $|\mathcal{Y}| = \sigma_d(\mathcal{T}) = |\xi|$, $\mathcal{Y}$ must be the required base.

**NEW SFM ALGORITHM**

Under the constraints, the structure-and-motion of the 3D wireframe is fully described by the depths of reciprocal depths of the vertices in the base $\mathcal{Y}$, i.e. by $\xi$, and by the six motion parameters $V$ and $\Omega$. However, the constraints have done nothing to resolve the depth/speed scaling ambiguity, and so we must still reduce the number of the parameters by one to $|\xi| + 5$.

Here, we fix the reciprocal depth of one of these independent vertices, say $\xi_1$, to unity.

The SFM problem becomes one of minimizing:

$$D(p) = \sum_{v} w_v (|v_1| - |v_{eod}|)^2$$  \hspace{1cm} (27)$$

subject now only to the inequality conditions:

$$B's' = B'H\xi > 0$$  \hspace{1cm} (28)$$

Here, the parameter vector is $p = (\xi_2 \ldots |\xi_d|, V, \Omega)'$, $B'$ is $B$ after column permutation, and $H$ is a linear transformation.

The complete procedure to obtain structure from motion is:

1. Label the line drawing or 2D vertex-edge graph.
2. Find the maximal position-free incidence structure $S^*$ using theorem 1.
3. Find the maximal independent set of vertices and thereby which vertices are associated with $\xi$.
4. Set $\xi_1 = 1$ and guess initial values for the parameters $(\xi_2 \ldots |\xi_d|)$ that satisfy $B'H\xi > 0$ and guess initial values for the six motion parameters $V$ and $\Omega$.
5. Starting with these initial values, minimize $D$ with respect to the parameters. If $D_{min}$ is below some threshold, and the $\xi$ at minimum satisfies the inequalities, go to step 6. Otherwise, go to step 4 and use a new guessed starting point.
6. If $\mathcal{R}^* = \mathcal{R}$, end. Otherwise, if $\mathcal{R}^* \neq \mathcal{R}$ the scene positions might not satisfy the constraints in $\mathcal{R} - \mathcal{R}^*$ because these have been removed. Correct the positions of the vertices involved with elements in $\mathcal{R} - \mathcal{R}^*$ by finding the intersections of the surfaces already computed. Then end.

**EXPERIMENTS**

**Toy truck**

Figure 5 shows the line labelling derived from the segmentation of the truck.

The entire incidence structure proves to be position-free in this case, and the base set contains five independent vertices which are used in the SFM optimization. Thus, the dimensionality of the optimization is reduced from 21 to 10. The reconstruction with constraints is shown in Figure 6a, and that without in Figure 6b. There is a substantial improvement in the recovered structure, particularly marked for this case of translation towards the camera because there is very little depth information around the focus of expansion, here at the image centre.

**Figure 5. Line labelling for the truck**
As well as reducing the dimensionality of the problem, we have also recovered explicitly the planar faces of the object. Recall that the first $3m$ components of $\eta$ contain the surface normals of the planar faces of the reconstructed object. Murray\textsuperscript{6} has described a method of matching surface normal and relative position data to CAD-type models. The method is based on that of Grimson and Lozano-Pérez\textsuperscript{4}, but develops geometrical matching constraints appropriate when the overall scale of the 3D data is unknown. This is the case here, because the structure data still suffer the depth/speed scaling ambiguity.

A data-to-model match is grown by considering the compatibility of the following metrics between pairs of data patches ($a$ and $b$) and pairs of model faces ($i$ and $j$):

$$\begin{align*}
\text{Data} & : \mathbf{N}_a, \mathbf{N}_b, \mathbf{d}_{ab}, \mathbf{d}_{ab}, \mathbf{N}_{ab}, \mathbf{D}_{ab}; \\
\text{Model} & : \mathbf{n}_i, \mathbf{n}_j, \mathbf{d}_i, \mathbf{d}_j, \mathbf{n}_{ij}, \mathbf{d}_{ij};
\end{align*}$$

The vector $\mathbf{N}_a$ is the unit normal to data patch $a$, $\mathbf{D}_{ab}$ is the unit vector in the direction between patches $a$ and $b$, and $\mathbf{N}_{ab} = \mathbf{N}_a \times \mathbf{N}_b$, and similarly for the model metrics. The various vectors are illustrated in Figure 7. Because the data normals have sensing errors, and

Table 1. Two interpretations feasible under the pairwise constraints. The second is found globally invalid by transformation

<table>
<thead>
<tr>
<th>Data patches</th>
<th>$1$</th>
<th>$2$</th>
<th>$3$</th>
<th>$4$</th>
<th>$5$</th>
<th>$6$</th>
<th>$7$</th>
<th>$8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model faces</td>
<td>$1$</td>
<td>$4$</td>
<td>$8$</td>
<td>$8$</td>
<td>$7$</td>
<td>$7$</td>
<td>$1$</td>
<td>$2$</td>
</tr>
<tr>
<td>Match 2</td>
<td>$4$</td>
<td>$9$</td>
<td>$9$</td>
<td>$7$</td>
<td>$7$</td>
<td>$1$</td>
<td>$2$</td>
<td>$3$</td>
</tr>
</tbody>
</table>

because the model faces have finite extent, both sets of metrics exhibit ranges of validity, which much overlap for consistency.

The surface normals from $\eta$ are normalized and placed at the centre of each reconstructed face, as shown in Figure 8a. Figure 8b shows the labelling of faces of the surface model. Because of symmetry, there are two matches which are feasible under the constraints, shown in Table 1.

A match which is feasible under the local pairwise constraints does not necessarily possess a valid global transformation $(R, t, F)$ relating model and sensor spaces, $\mu$ and $\sigma$, as $\sigma = FR\mu + t$, where $R$ is a rotation matrix, $t$ is a translation, and $F$ is a scaling factor. Using each feasible match we derive first the rotation $R$ (using the quaternion technique of Faugeras and Hébert\textsuperscript{11}), and then the translation and scaling that best relate model and data spaces. We then assess whether this represents a good global transformation by determining the overall deviation of the sensed patch positions from their respective matched faces after transformation.

In the case of the toy truck, this process enables us to reject the second feasible interpretation as globally invalid. The scale factor derived from the first feasible and globally valid interpretation finally resolves the depth/speed scaling ambiguity\textsuperscript{5}, enabling the recovery of absolute depths and translation velocity. For example, the vertical width of the toy truck was 76 mm, and that recovered was 71.1 mm; the vertical translational velocity was $V = (0, 0, -20)$ mm per frame, and that computed was $(0.3, 0.1, -17.9)$ mm per frame.

Chipped block
We include a second example with an incidence structure which is not position-free. Figure 9 shows the
