Recognising Polyhedral Objects from a Single Perspective View

K. C. Wong and J. Kittler

Dept. of Electronic and Electrical Engineering,

University of Surrey,

Guildford, Surrey GU2 5XH, United Kingdom

Abstract

This paper considers the problem of recognising 3D polyhedral objects from a single perspective image. A hypothesis-verification paradigm based on a local shape representation is presented. In the framework, 2D vertices interpreted as the projection of a trihedral vertex which is a 3D spatial vertex with three line emanating from the tip are employed as seed features for model invocation and hypothesis generation. To simplify the perspective analysis, Kanatani [7] has proposed an intuitive and elegant technique. Using the technique, we derive a fourth-degree polynomial for interpreting a trihedral vertex. The contribution of our solution is that there are no restrictions on angles between the vertex edges. To reduce the number of hypotheses generated from scene-model vertex assignments, and recover the complete object pose, we propose a composite feature, namely vertex-CS feature by combining a trihedral vertex and a V-junction which share a common edge. The geometric constraint of this composite feature is derived. A matching strategy used in the recognition system is discussed. The feasibility of the proposed method is illustrated on real data.

1 Introduction

In general, there are several distinct phases in model-based matching of rigid objects. Two off-line stages are model generation and model analysis. The former is required for constructing a CAD-like database of models. The latter is exploited to identify and organise model features into structures for matching and for developing strategies for the execution of the matching task. The two main run-time stages are hypothesis generation and verification. The first of these involves extracting interesting 2D geometric features from an image and then generating possible poses of scene objects using the geometric cues of the plausible model-scene correspondences. The subsequent object verification process is thus provided with tight constraints on where to search for confirmatory evidence of model existence. The model verification process performs a detailed check of the description of the projection of 3D features and 2D image data, confirming the feature presence and accounting for features which are not observed. Most of the existing recognition systems which use the above approach to accomplish the image interpretation task, rely on cues derived from the geometric relationships between model-scene correspondences [8], [3], [2].

In this paper, we describe a model-based polyhedral object recognition system for identifying the scene-model correspondences and estimating the poses of the scene object from a single perspective image. A hypothesis-verification paradigm based

BMVC 1992 doi:10.5244/C.6.61

on local shape properties is presented. In the framework, trihedral vertices and their composite with V-junctions are employed as key features for model invocation and hypothesis generation. There are several reasons for choosing this feature : the number of vertices extracted from a scene is generally manageable; they are robust in the presence of moderate noise; they are qualitative invariants over a wide range of view points [5]; they can constraint the transformation between the model and camera frames. Although our approach is inspired by Kanatani [7] the proposed method advances the state of the art in at least three important respects. Firstly, we demonstrate that the pose of a scene object can be recovered using a very intuitive formulation, by analytically solving a quartic equation derived from the geometric constraint of a model-scene vertex pair. Moreover, our method is not restricted to objects, with 3D vertices involving at least two right angles. Secondly, we have derived the geometric relationship of a composite feature formed by a vertex and a V-junction to reduced the search space and recover the translation of the scene object accurately. Thirdly, the process of estimating the pose of a scene object is broken down into two stages whereby in the first stage no quantitative information is required about edge length. Many false hypotheses can be pruned out at this first stage without the need for full edge visibility. This greatly simplifies the problem of verification.

Many researchers have attempted to solve the problem of determining the pose of a spatial 3D vertex from its orthographic [5], [6] or perspective [1], [4], [7] projection. Among these approaches, we find that the formulation and framework proposed by Kanatani [7] are most elegant and intuitive. However, the analytical solution derived by him can only deal with a corner with a minimum of two right angles out of three.

2 Solving vertex edge orientations

To determine the relative pose of a scene object with respect a camera frame, we first concentrate on solving the edge orientations of a spatial vertex measured in the camera frame. The formulation and analysis of 3D geometric primitive features being

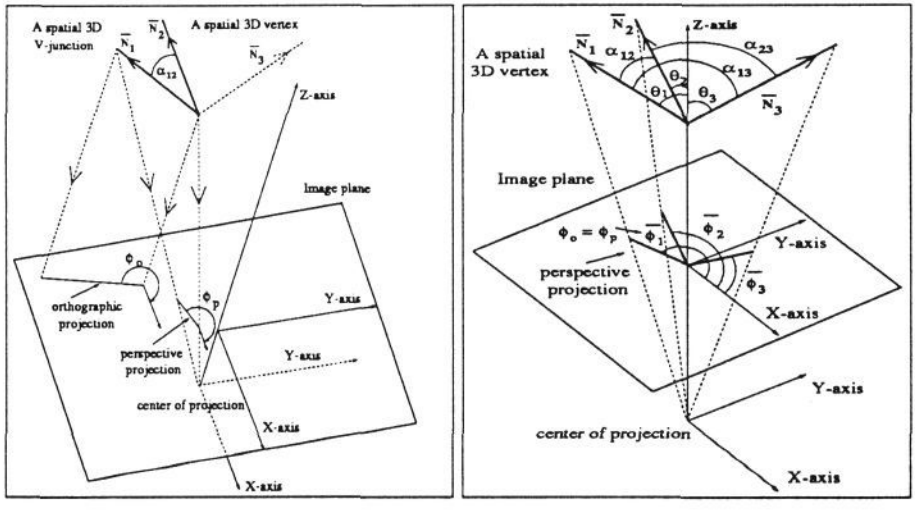

Figure 1: (a) A scene vertex being viewed in general position. (b) A configuration of a vertex transformed to a canonical position

viewed under perspective projection in general position is very complicated. Hence, the geometric meaning of the equations derived from these formulations is in general implicit and non-intuitive. To resolve the perspective geometry problem, Kanatani [7] proposed a technique to move scene features to be analysed into an image origin, namely the canonical position where the analysis of the scene feature can be greatly simplified. In particular, the difficulty of understanding the perspective projection of a spatial V-junction can be reduced significantly as the perspective effect is reduced to orthographic in the canonical position. For example in Fig. 1(a), the image angles of the edges of a spatial V-junction observed under perspective (ϕ_p) and orthographic (ϕ_o) projection are different in general position. However, the distinction between these two disappears ($\phi_p = \phi_o$) in the canonical position where the optical axis of the camera intersect the tip of the V-junction (see Fig. 1(b)). Therefore, the orientation of the edge $\bar{N}_1 = \Re^t N_1$ of a V-junction can be simply expressed as $\bar{N}_1 = \sin\theta_1 \cos\bar{\phi}_1 \bar{i} + \sin\theta_1 \sin\bar{\phi}_1 \bar{j} + \cos\theta_1 \bar{k}$, where θ_1 is the angle between the edge $\bar{N}_1$ and an optical axis of a camera and $\bar{\phi}_1$ is an orientation of the edge under *perspective or orthographic projection*. $\Re$ is a *standard transformation* which maps an image point to the origin (0,0) (see [7] for more details).

Having expressed the edge orientations of a V-junction, the 3D true angle α_{12} between the edges $\bar{N}_1$ and $\bar{N}_2$ shown in Fig. 1(b) can easily be written as $\bar{N}_1 \cdot \bar{N}_2 = \cos\alpha_{12}$. Similarly, a system of trigonometric equations can be derived in the same manner for a spatial vertex shown in Fig. 1(b). Replacing the $\sin\theta_i$ and $\cos\theta_i$ in the resultant equations with $\frac{2t_i}{1+t_i^2}$ and $\frac{1-t_i^2}{1+t_i^2}$ respectively, where $t_i = \tan\frac{\theta_i}{2}$, for $(i = 1, 2, 3)$, we find

$$(k_1 - 1)\, t_1^2\, t_2^2 + (k_1 + 1)\,(t_1^2 + t_2^2) - 4\, q_1\, t_1\, t_2 + (k_1 - 1) = 0 \tag{1}$$

$$(k_2 - 1)\, t_2^2\, t_3^2 + (k_2 + 1)\,(t_2^2 + t_3^2) - 4\, q_2\, t_2\, t_3 + (k_2 - 1) = 0 \tag{2}$$

$$(k_3 - 1)\, t_1^2\, t_3^2 + (k_3 + 1)\,(t_1^2 + t_3^2) - 4\, q_3\, t_1\, t_3 + (k_3 - 1) = 0 \tag{3}$$

where $k_i = \cos\alpha_{i,(i \bmod 3)+1}$ and $q_i = \cos(\phi_i - \phi_{(i \bmod 3)+1})$. From Eq.(3), we find that

$$t_3^2 = \frac{4\, q_3\, t_1\, t_3 - (k_3 + 1)\, t_1^2 - k_3 + 1}{(k_3 - 1)\, t_1^2 + k_3 + 1} \tag{4}$$

Substituting this expression for t_3^2 into Eq.(2) gives,

$$t_3 = \frac{(k_2 - k_3)\, t_1^2\, t_2^2 + (k_3 + k_2)\, t_1^2 - (k_3 + k_2)\, t_2^2 + k_3 - k_2}{2\,(q_3\,((k_2 - 1)\, t_2^2 + (k_2 + 1))\, t_1 + q_2\,((1 - k_3)\, t_1^2 - (k_3 + 1))\, t_2)} \tag{5}$$

Substituting back the expression of t_3 into Eq.(3) yields,

$$\delta_4\, t_2^4 + \delta_3\, t_2^3 + \delta_2\, t_2^2 + \delta_1\, t_2 + \delta_0 = 0 \tag{6}$$

where,

$$\begin{aligned}
\delta_4 &= (k_2 - k_3)^2\, t_1^4 + 2\,((k_3^2 - k_2^2) + 2\, q_3^2\,(k_2^2 - 1))\, t_1^2 + (k_2 + k_3)^2 \\
\delta_3 &= 8\, q_2\, q_3\,((1 - k_2\, k_3)\, t_1^2 - (1 + k_2\, k_3))\, t_1 \\
\delta_2 &= 2\,(2\, q_2^2\,(k_3^2 - 1) + (k_2^2 - k_3^2))\,(t_1^4 + 1) \\
&\quad + 4\,(2\,(q_2^2\,(k_3^2 + 1) + q_3^2\,(k_2^2 + 1)) - (k_3^2 + k_2^2))\, t_1^2 \\
\delta_1 &= 8\, q_2\, q_3\,((1 - k_2\, k_3) - (1 + k_2\, k_3)\, t_1^2)\, t_1 \\
\delta_0 &= (k_2 + k_3)^2\, t_1^4 + 2\,((k_3^2 - k_2^2) + 2\, q_3^2\,(k_2^2 - 1))\, t_1^2 + (k_2 - k_3)^2
\end{aligned}$$

Equation (1) can be rewritten,

$$\rho_2\, t_2^2 + \rho_1\, t_2 + \rho_0 = 0 \tag{7}$$

where, $\rho_2 = (k_1 - 1)\, t_1^2 + (k_1 + 1)$; $\rho_1 = -4\, q_1\, t_1$; $\rho_0 = (k_1 + 1)\, t_1^2 + (k_1 - 1)$; and consequently from eq. (7), we get $t_2^2 = -\frac{\rho_1\, t_2 + \rho_0}{\rho_2}$. Substituting this expression for t_2^2 into Eq.(6) yields,

$$t_2 = \frac{(\rho_1^2 - \rho_0\, \rho_2)\, \rho_0\, \delta_4 - \rho_0\, \rho_1\, \rho_2\, \delta_3 + \rho_0\, \rho_2^2\, \delta_2 - \rho_2^3\, \delta_0}{(2\, \rho_0\, \rho_2 - \rho_1^2)\, \rho_1\, \delta_4 + (\rho_1^2 - \rho_0\, \rho_2)\, \rho_2\, \delta_3 - \rho_1\, \rho_2^2\, \delta_2 + \rho_2^3\, \delta_1} \tag{8}$$

Substituting this expression for t_2 into Eq.(7) we obtain,

$$\begin{aligned}
&(((\rho_1^2 - 2\, \rho_0\, \rho_2)^2 - 2\, \rho_0^2\, \rho_2^2)\, \delta_0 + (3\, \rho_0\, \rho_2 - \rho_1^2)\, \rho_0\, \rho_1\, \delta_1 \\
&+(\rho_1^2 - 2\, \rho_0\, \rho_2)\, \rho_0^2\, \delta_2 - \rho_0^3\, \rho_1\, \delta_3)\, \delta_4 + ((3\, \rho_0\, \rho_2 - \rho_1^2)\, \rho_2\, \rho_1\, \delta_0 \\
&+(\rho_1^2 - 2\, \rho_0\, \rho_2)\, \rho_2\, \rho_0\, \delta_1 - \rho_0^2\, \rho_1\, \rho_2\, \delta_2)\, \delta_3 \\
&+((\rho_1^2 - 2\, \rho_0\, \rho_2)\, \rho_2^2\, \delta_0 - \rho_0\, \rho_1\, \rho_2^2\, \delta_1)\, \delta_2 - \rho_1\, \rho_2^3\, \delta_0\, \delta_1 \\
&+\rho_2^4\, \delta_0^2 + \rho_0\, \rho_2^3\, \delta_1^2 + \rho_0^2\, \rho_2^2\, \delta_2^2 + \rho_0^3\, \rho_2\, \delta_3^2 + \rho_0^4\, \delta_4^2 = 0
\end{aligned} \tag{9}$$

Substituting the known constants $\delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \rho_0, \rho_1$ and ρ_2 into Eq.(9), we obtain a polynomial equation of degree 16 with no odd terms in one unknown t_1. Replacing t_1^2 with the trigonometrical $\frac{1-\cos(\theta_1)}{1+\cos(\theta_1)}$ yields the following fourth-degree polynomial in one unknown $\cos^2(\theta_1)$:

$$A_4\, \cos^8(\theta_1) + A_3 \cos^6(\theta_1) + A_2 \cos^4(\theta_1) + A_1 \cos^2(\theta_1) + A_0 = 0 \tag{10}$$

This quartic equation can be solved analytically or using iterative numerical method. t_2 and t_3 can be obtained by substituting t_1 into Eq.(1) and Eq.(3) respectively. All the solutions of t_1, t_2 and t_3 are then verified using the Eq.(2). The angles between the trihedral vertex edges can be easily determined from the *t-formula*, $\theta_i = 2\, \tan^{-1}(t_i)$, where $0 < \theta_i < \pi$. Many hypotheses can be pruned away during the recovery of the edge orientations of a vertex leaving very few hypotheses to be verified. Having determined the vertex edge orientations, we will describe the pose determination problem in next section.

3 Pose Estimation

Formally, the problem of pose estimation may be defined as follows : *Given a set of N three dimensional vectors with respect to an inherent object model coordinate system, and the 2D perspective projection of the corresponding N vectors of a scene object with respect to a camera frame, estimate the relative rotation R_{MC} and the translation vector T_{MC} to define the relationship between the object model and camera frame.* To solve this problem, we first concentrate on solving the relative rotation transform consisting of three degrees of freedom. The translation vector can easily be recovered once the former is determined. Consider the perspective geometry of a camera model depicted in Fig. 2(a), the image plane is assumed to be in front of the center of projection so as to acquire an upright scene image. The focal length, **foc** is the normal distance from the center of projection to the image plane. Based on the above configuration, the position of the scene vertex P_s can be expressed in a camera frame centered at the origin **E** as $P_s = R_{MC}\, P_w + T_{MC}$, where R_{MC} is the relative orientation between the model and camera frame and T_{MC} is a translation vector. In the next subsections, we will describe the methods for computing these parameters.

3.1 Relative Rotation

To determine the relative rotation, we decompose the rotation transform into model-to-vertex R_{MV} and camera-to-vertex R_{CV} transforms. Consider the edges of a trihedral vertex E_i, $i = 1,2,3$, described with respect to an object model coordinate system. An orthogonal vertex-based frame can be constructed by using the Gram-Schmit orthogonalization process, $M_i' = E_i - \sum_{j=1}^{i-1} \frac{E_i \cdot M_j'}{\|M_j'\|^2} M_j'$, and then normalised to obtain unit vectors $M_i = \frac{M_i'}{\|M_i'\|}$. The transformation R_{MV} of vertices P_w with respect to an object model coordinate system to vertices P_v with respect to a vertex-based coordinate system (see Fig. 2(a)) can be expressed as $P_v = R_{MV} P_w$ where $R_{MV} = \begin{pmatrix} m_{1x} & m_{1y} & m_{1z} \\ m_{2x} & m_{2y} & m_{2z} \\ m_{3x} & m_{3y} & m_{3z} \end{pmatrix}$. Having determined the edge orientations of a vertex in the canonical position, the orientations of the edges in the scene can be recovered by $N_i = \Re \bar{N}_i$, $i = 1,2,3$, which are described with respect to the camera coordinate system. An orthogonal vertex-based frame is constructed using the corresponding recovered edges by taking $C_i' = N_i - \sum_{j=1}^{i-1} \frac{N_i \cdot C_j'}{\|C_j'\|^2} C_j'$, and then normalised to obtain unit vectors $C_i = \frac{C_i'}{\|C_i'\|}$. The transformation R_{CV} of vertices P_s' with respect to a camera coordinate system, which is centered at the origin of the world coordinate system $\mathbf{O}$, to vertices P_v with respect to a vertex-based coordinate system can be expressed as $P_v = R_{CV}\ P_s'$ where $R_{CV} = \begin{pmatrix} c_{1x} & c_{1y} & c_{1z} \\ c_{2x} & c_{2y} & c_{2z} \\ c_{3x} & c_{3y} & c_{3z} \end{pmatrix}$. Having determined R_{MV} and R_{CV}, the rotation transform which maps an object model P_w to the scene feature point P_s' with respect to the camera coordinate system, which is centered at the origin $\mathbf{O}$ of the model frame can be written as $P_s' = R_{MC}\ P_w$ where $R_{MC} = R_{CV}^t \times R_{MV}$.

3.2 Translation

To determine the translation from an object model to a scene, one of the three line segments of an image vertex must be the projection of the full edge of a spatial 3D vertex. We will defer a detailed description of this issue to Section 4. Here we shall assume that one of the line segments of length l_i is the true projection of an edge of length L_i of a spatial vertex in canonical position. The orientation of the edge orientation θ_i can be evaluated from the solution derived in Section 2. The positional vector D_i of the tip of the vertex in canonical position can be easily expressed as $D_i = 0\ \tilde{i} + 0\ \tilde{j} + L_i\ (\frac{foc}{l_i} \sin\theta_i - \cos\theta_i)\tilde{k}$, where foc is the focal length of the camera. Let the positional vector of the tip of the corresponding model vertex be D_m. The translation from the model to the scene can then be computed using, $T_{MC} = \Re\ D_i - R_{MC}\ D_m$ Having determined the complete pose of the scene object, the results can be employed for predicting the description of the the 2D scene features in the verification phase of the recognition system. In the next section, we will introduce a feature primitive which can be reliably used for computing the translation vector. Furthermore, the geometric constraint of the composite feature will be derived.

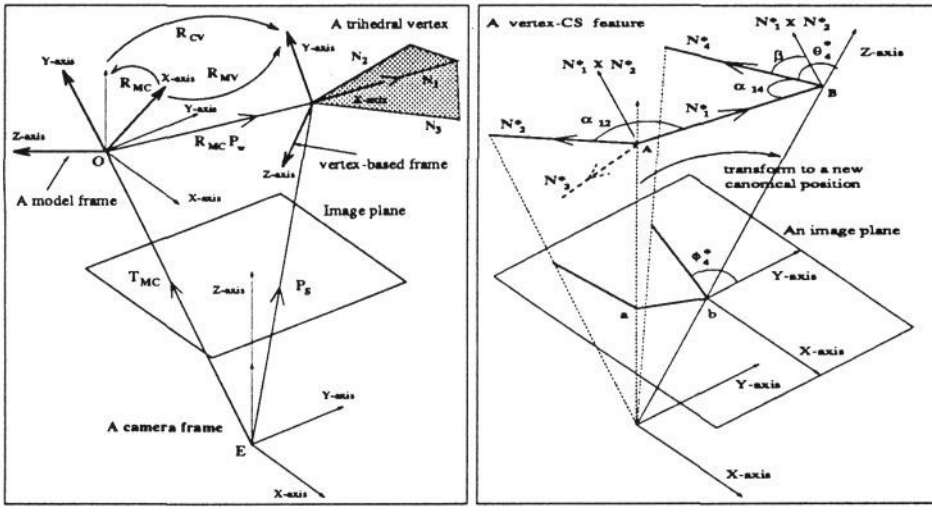

Figure 2: (a) A simplified pin-hole camera. (b) A composite vertex-CS feature

4 Composite Vertex-CS feature

To determine the translation from an object model to a scene, one of the three line segments of an image vertex must be the projection of the full edge of a spatial 3D vertex. In order to select at least one plausible line segment of a vertex extracted from a real image, we introduce a primitive, namely vertex-CS feature, by combining a vertex and a V-junction which share a common line segment as shown in Fig. 2(b). This feature description can be reliably extracted from image data. The common line **ab** of the vertex-CS feature is taken as the true projected 3D edge **AB** as the V-junctions **a** and **b** are most simply interpreted as the projections of the two spatial vertices **A** and **B** respectively. The geometric constraint imposed by the vertex-CS feature can be employed to discard the inconsistent hypotheses generated from the model-scene vertex pairs.

Now, we shall derive the geometric constraint of the composite vertex-CS feature. The edge orientation $\bar{N}_1, \bar{N}_2$ and $\bar{N}_3$ of the vertex at the canonical position **a** can be determined by solving the equations described in Section 2. All the edge orientations N_i^* are represented as unit vectors. To analyse the V-junction **B** of the composite feature, the camera is transformed to a new canonical position by pointing the optical axis to the vertex **B**. In another words, transforming the V-junction **b** to the origin of the image plane. Let $\Re_{ab}$ be the *standard transform* which maps the image point **b** to **a**. The edge orientations N_1^*, N_2^* and N_3^* under the new canonical position **B** can then be written as $(\Re_{ab}^t \bar{N}_1), (\Re_{ab}^t \bar{N}_2)$ and $(\Re_{ab}^t \bar{N}_2)$ respectively, where t denotes a transpose. The angle α_{14} of the vertex can be expressed as a dot product $N_1^* \cdot N_4^* = \cos(\pi - \alpha_{14})$. After some manipulation, the angle θ_4^* between the edge N_4^* and the optical axis can be expressed as :

$$\theta_4^* = \cos^{-1} \frac{\cos(\pi - \alpha_{14})}{\sqrt{\mu^2 + (n_{1z}^*)^2}} + \tan^{-1} \frac{\mu}{n_{1z}^*} \tag{11}$$

Where $\mu = n_{1x}^* \cos\phi_4^* + n_{1y}^* \sin\phi_4^*$. The edge angle θ_4^* can be determined by substitut-

ing the projected edge angle ϕ_4^* and the true 3D angle of the vertex **B** into the Eq.(11). The solutions can then be substituted into the equation, $\beta = \cos^{-1}(\frac{(N_1^* \times N_2^*)}{|N_1^* \times N_2^*|} \cdot N_4^*)$, which is the angle between the normal $N_1^* \times N_2^*$ and the edge N_4^*. The measured angle β should correspond to the pre-computed angle β of the hypothesised model. The model-scene vertex pair hypotheses which agree with angle β will be considered in the verification. It is worth noting that the geometric constraint imposed by the composite feature does not require quantitative information about the edge length. In the next section, we will discuss the matching strategies used for recognising polyhedral objects from a single perspective image.

5 Matching Strategy

Several modules are integrated into our system to accomplish the task of polyhedral object recognition. They can be briefly described as follows. Vertices extracted from the image satisfy some predefined criteria such as junction region size, the length of the radiating line segments and the angles between them. In order to control the combinatorial explosion associated with unconstrained association of model-scene vertex pair assignments, high quality vertices with small region size, relatively long segments and reasonable angle sizes between them are extracted from the given scene first. All 3 possible combinatorial assignments of corresponding edges between a model and image vertex are considered. Scene vertices which match the geometric configuration of at least one vertex stored in the model base will be considered in the subsequent process. The model with at least one vertex satisfying the geometric constraints will be registered as a consistent interpretation.

The remaining vertex candidates will be employed to provide a tight constraint on where to search for V-junctions which share one of three line segments of the feasible vertices. When searching for plausible V-junctions to form feasible composite features, scene vertices were processed in the descending order of their line length. If no plausible composite features are found, we can then relax the threshold on the junction region size of the V-junctions. In the very worst case, we may treat those T-junctions when one of the vertex line segments is either the cap or the bar, as required V-junction. The pose of hypothesised object models is then estimated using the geometric relationships derived from the model and scene composite features. A simple visibility test is performed on the residual hypotheses. A trihedral vertex which is interpreted as a scene vertex should contain at least two visible surfaces otherwise the associated hypothesised model can be removed from the candidate list for further consideration.

The 2D description of each backprojected model is compared with the features extracted from the image. First, we count the number of 2D junctions of the hypothesised objects that overlap a junction extracted from the scene image. Two junctions are said to be overlapping if they are within a proximity threshold value and their angles and orientation match to an allowable tolerance. After comparing every projected junction of the hypothesised objects with the 2D junctions extracted from the scene, the hypotheses with the greatest number of matched junctions will be invocaked. The aim of this stage is to select the hypothesis with the greatest number of features overlapping with the scene data. To achieve this, the nearest scene line from each projected 2D model line length is identified. If it is within an allowable threshold, then the line length is divided by the corresponding projected line length. These computed quotients are then summed up and divided by the number of visible projected edges of the hypothesised model yielding the confidence measure for each hypothesis.

6 Experimental Results

Real images were employed to test the reliability, robustness and computational efficiency of the polyhedral recognition system described in this paper. The model and camera frames employed in all the experiments are designated as right-handed coordinate system. There are two pyramid models and a roof model used in this experiment (see Fig. 3(a)). The images were taken with a standard CCD camera. Fig. 3(b) and (c) show the grey-level image and lines extracted by Hough process, respectively. In this example, 15 vertices identified from the scene are shown in Fig. 3(c) marked **Tn**. Some of these vertices, were generated by extraneous lines due to effects such as shadowing. All the vertices of each model were compared exhaustively with each vertex extracted from the test scene. The number of admissible solutions generated from matching the pyramid model #1, #2 and the roof model #3 against all the scene vertices are 243, 256 and 313, respectively. In some cases, there were no feasible solutions found when establishing the geometrical relationships between individual vertices of the pyramid models and the scene vertices. For hypothesised candidates which do satisfy the geometrical constraint of a model-scene vertex pair, the grouping process generates feasible composite features around the line segments of the scene vertices. In this example, 59 vertex-CS composite features were extracted from the scene. These composite features were checked using the geometric constraint.

Hypotheses remaining after applying the vertex-CS geometric constraint and simple visibility test for the model #1, #2 and #3 were reduced by about 55.6%, 57.8% and 66.8%, respectively. Using the information of the vertex-CS composite feature, the poses of all the admissible hypotheses were computed and their 2D predictions were compared with the geometric primitives such as line segments and V-junctions extracted from the scene shown in Fig. 3(c). Both the correct and wrong candidates were generated and their confidence measures or the close correspondence between backprojected model lines and lines extracted from a given scene image were computed. Some of the incorrect hypotheses generated from the pyramid and roof model-scene vertex assignments are shown in Fig. 3(d) and (e) respectively.

In this instance, the pyramid model #2 and the roof model #3 matched the scene object #2 and #5 correctly. The number of matched junctions in each case are 5 (out of 6) and 6 (out of 7) respectively. The confidence measures of both cases are are 90.2% and 86.8%. Unfortunately, in the case of computing the confidence measures for the hypotheses generated from the matching of model pyramid #1 against scene vertices, the most plausible candidate among the admissible solutions generated by matching against scene vertex T_2 was 5.6% lower than the best hypothesis (81.7%) generated from matching model #1 against the scene vertex T_8. Many experiments were performed using the test scene containing the two pyramid models. The two pyramid models always matched to the scene pyramid with better quality 2D features. This was due to the fact that the 2D descriptions of the two pyramid models under perspective projection were very similar. Futhermore, the 2D description of the scene object #1 was *degraded significantly relative* to the scene object #2. In this case, the correct hypothesis can only be found if the distance from the camera to the *table top* is known a priori. The computed furthest distance for the five hypothesised pyramid models for object #1 at the top of the list differed from the edge of the table by a factor of two. We acknowledge that in general the distance to the table from the the camera may not be known in advance. However, the correct model for the scene pyramid object #1 could only be identified by making this extra assumption. Fig. 3(f) shows the superimposed models onto the scene objects using the computed transformation.

Next, the proposed method was explored on a cluttered scene shown in Fig. 4(a).

The target objects are the pyramid and the roof model. They are labelled with **S1** and **S2** respectively (see Fig. 4(a)). One of the two visible trihedral vertices of the roof model was occluded by a "computer mouse". There are 8 vertices extracted from this scene (see Fig. 4(b)). The numbers of hypotheses generated by matching pyramid #2 and roof model #3 vertices against the scene vertices were 84 and 119 respectively. Some of the incorrect hypotheses generated from the roof model-scene vertex assignments are shown in Fig. 4(c). After applying vertex-CS constraints, the number of hypotheses for each case were reduced by 42.9% and 68.1%, respectively. The correct models for the scene objects were identified. The confidence measures for each case were 72.6% and 66.1%. The superimposed models onto the scene objects using the computed transformation are shown in Fig. 4(d).

7 Conclusion

In this paper, we have presented a paradigm based on local shape properties for identifying the scene-model correspondences and estimating the poses of the scene object from a single perspective image. In the framework, vertices and composite vertex-CS feature are employed as seed features for model invocation and hypothesis generation. We have derived an analytical quartic equation for describing geometric relationships of a model-scene vertex pairs, with not restriction on the angles between vertex edges. We have introduced a vertex-CS feature of which the effectiveness and the geometric constraint are presented. Using the seed features, Many false hypotheses can be pruned away without concerns about full edge visibility which greatly simplifies the problem of computational intensive verification process. The experimental results reported confirm the feasibility of the proposed paradigm.

Clearly, the robustness of the method depends entirely on the extraction of the vertices and composite features. Some features may however be missing due either to occlusion or inadequate low-level processing. To cope with these problems, the low confidence or poor quality features can be enhanced by modifying the thresholds on proximity and orientation checks. However, if the tolerance is too large, the number of features extracted from a scene may cause the model-scene correspondences to grow exponentially. This is one of the important issues which can be solved by developing an adaptive control mechanism for providing an interactive environment between the matching phase and low-level or feature grouping process.

Acknowledgement

This work was carried out as part of the **ESPRIT** Basic Research Action Project BR3038, "Vision as Process". KCW is supported by an (UK) **ORS** Award.

References

[1] Stephen T. Barnard, "Interpreting Perspective Images", *Artificial Intelligence*, **21**, pp 435-462, 1983.

[2] M. Dhome, M. Richetin, J. T. Lapresté and G. Rives, "Determination of Attitude of 3-D Objects from a Single Perspective View", *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. 11, No. 12, Dec 1989.

[3] R. Horaud, "New Methods for Matching 3-D Objects with Single Perspective Views", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-9, No. 3, pp 401-412, May 1987.

[4] R. Horaud, B. Conio, O. Leboulleux and B. Lacolle, "An Analytic Solution for the Perspective 4-Point Problem, *CVGIP 47*, pp 33-44, 1989.

[5] T. Kanade, "Recovery of the three-dimensional shape of an object from a single view" *Artificial Intelligence*, 17, pp 409-460, 1981.

[6] K. Kanatani, "The Constraints on Images of Rectangular Polyhedra", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-8, No. 4, pp 456-463, 1986.

[7] K. Kanatani, "Constraints on Length and Angle", *CVGIP 41*, pp 28-42, 1988.

[8] D. G. Lowe, "Three-Dimensional Object Recognition from Single-Two Dimensional Images", *Artificial Intelligence*, 31, pp 355-395, 1987.

[9] T. Shakunaga and H. Kaneko, "Perspective Angle Transform : Principle of Shape from Angles" Int. Journal of Compute Vision, 3, pp 239-254, 1989.

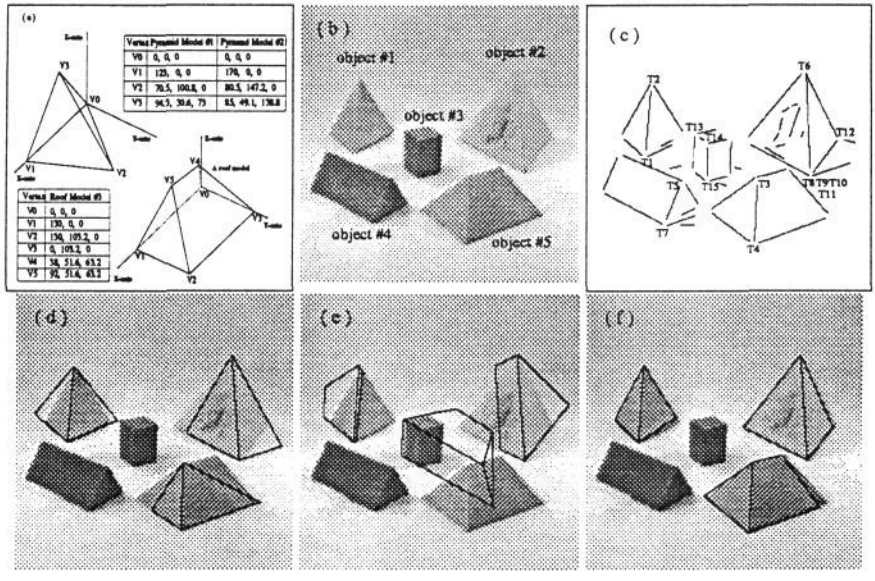

Figure 3: (a), (b), (c), (d), (e) and (f) *see text*

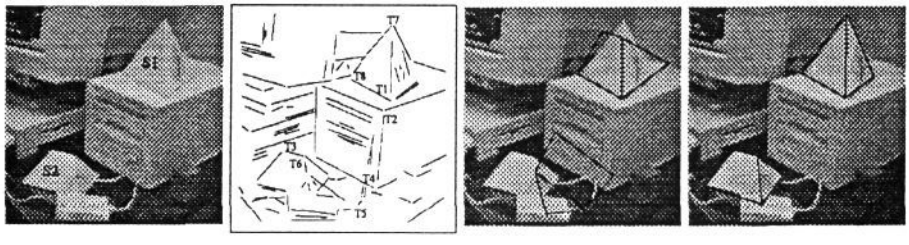

Figure 4: (a), (b), (c) and (d) *see text*