Active 3D Object Recognition using 3D Affine Invariants

Sven Vinther and Roberto Cipolla

Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, England

Abstract. We evaluate the power of 3D affine invariants in an object recognition scheme. These invariants are actively estimated by Kalman filtering the data obtained from real-time tracking of image features through a sequence of images. Object information is stored and retrieved in a hash table using the invariants as stable indices. Recognition takes place when significant evidence for a particular shape has been found from the table. Results with real data are presented, and the noise problems arising due to the *weak* perspective approximation and corner localisation errors are discussed. Preliminary results for extending this method to multiple object recognition in cluttered scenes are also presented.

1 Introduction

Central to any object recognition scheme are issues of model representation, feature extraction and feature matching. An extensive survey describing current methods of object representation and matching can by found in [1]. A number of different working recognition schemes have been proposed for simple 3D objects. Lowe [2] presents a 3D object recognition system which uses a single image, *perceptual groupings* and viewpoint consistency constraints to detect 3D objects from 2D data. Alignment [3] uses a minimal set of features to calculate the model-to-image transformation, which is then verified using back-projection of model edges. Other approaches make use of 3D data extracted with a range finder or computed from multiple calibrated views. An example is the approach by Grimson and Lozano-Perez [4] which operates by examining all hypotheses between segmented range data and model surfaces, and efficiently discarding inconsistent ones by using local constraints. An alternative approach involves *geometric invariants* [5].

2 Geometric Invariants

Geometric invariants are properties of image features which do not change under a variety of transformations. Those properties which are invariant under rigid 3D transformations and perspective or *weak* perspective projection, are particularly attractive in computer vision tasks such as 3D object recognition, since they allow an efficient object representation which can be used in shape indexing. A well-known example of invariance under perspective projection is the *cross-ratio* of four points on a line. In this paper we evaluate the power of 3D affine invariants in a 3D object recognition scheme.

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

Weak Perspective Throughout this paper the *weak* perspective assumption will be made about the projection of the 3D object into the image plane. *Weak* perspective is a good approximation of full perspective when the distance from object to camera, is much greater than the extent of the object (a ratio greater than 10:1) [6]. The use of this approximation means the transformation of a 3D object from world coordinates (X_w, Y_w, Z_w) to image points (u, v), which consists of a combination of rigid motions and the *weak* perspective projection, can be represented by the general 3D affine transformation matrix shown in (1).

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ 0 & 0 & 0 & t_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

2.1 3D Affine Invariants

The objects to be recognised are represented by point sets denoting the position of the object vertices. By creating a 3D basis (figure 1) using four of these points, it is possible to define the position of any other point $\mathbf{X_4}$ by a linear combination of the basis vectors with the appropriate coefficients (2).

$$\mathbf{X_4} = \mathbf{X_0} + \alpha\mathbf{E_1} + \beta\mathbf{E_2} + \gamma\mathbf{E_3} \tag{2}$$

These coefficients (α, β, γ) are in fact invariant under any 3D affine transformation (1) [7, 8], and we will refer to them as 3D *affine* invariants. The affine invariants contain a description of the 3D structure of the object, up to a 3D affine transformation. Since the effect of the intrinsic camera parameters on projection can be absorbed into the affine transformation, affine invariants are unchanged by these aswell, with the benefit to the recognition system that no extensive calibration is required. The recovery of the 3D affine invariants from an unconstrained 2D image is not possible [9], unless some external prior information about the model is available (*e.g.* shape symmetry [10]). However the extraction of the 3D affine invariants becomes fairly simple from multiple images with known point correspondence [5, 11]. In this paper we estimate the affine invariants by Kalman filtering the data obtained from real-time tracking of image features through a sequence of images. This has the advantages of reducing the uncertainty in the invariant estimate and it avoids the correspondence problem present with stereo image pairs.

3 Object Model Acquisition and Recognition

A preliminary task for any object recognition system is the reliable extraction of features from raw intensity images. We wish to detect corners in the image which relate to the projections of object vertices (as in [3]). These vertices are tracked through a sequence of images, and a Kalman Filter is used to optimally estimate the invariants from the stream of data. The feature extraction and tracking processes are described in greater detail in [11]. The next task is to match the invariants extracted from the image data with those computed from

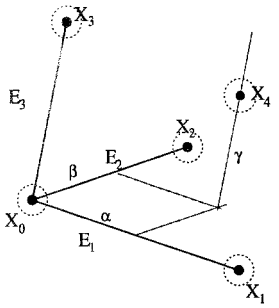


Fig. 1. Affine Basis.

a 3D model. Instead of calculating all the invariants for each 3D object model at recognition time, the information can be precomputed and stored in a database off-line. This greatly speeds up the final recognition process at the cost of extra storage space. Geometric hashing [13] is used to implement this data storage.

3.1 Optimal Estimation of the Invariants and their Uncertainty

The most important stage in the recognition scheme is the accurate estimation of the 3D affine invariants. This goal is achieved by using a Kalman filter to integrate data from multiple views, and optimally estimate the invariant coefficients α, β, γ. In addition the Kalman filter returns the uncertainty of the invariant estimates, in terms of estimate variances.

Estimate Uncertainty There are a number of factors which will affect the uncertainty of the 3D affine invariant estimate. Firstly the amount of noise expected in the track data needs to be considered. This will depend on the accuracy of point localisation by the corner detector, and will probably vary in different imaging environments. Its value can either be measured from test data or estimated, and input into the observation noise matrix of the Kalman Filter.

Secondly the 3D basis geometry will affect the sensitivity of the invariants to noise. If the basis is formed from 3D points which are nearly coplanar the resulting equations will become ill-conditioned, and thus provide noise sensitive estimates. The object rotation between the initial and final image will also affect how well-conditioned the equations are, in a similar fashion to the stereo camera baseline. The effect of the basis geometry is reflected in the variance returned by the Kalman Filter, and during recognition it may be necessary to try several different bases in order to find one which gives well-conditioned equations. This can be implemented by running several Kalman filters on the track data in parallel, each using a different set of points as the basis and selecting the invariants from the one with least uncertainty.

Finally some perspective distortion of the data will also occur, although we have assumed none is present in the computation of the affine invariants. The effect of this distortion is nonlinear, and it depends on the object pose relative to the camera, which is unknown. The size of this error will depend on the accuracy of the weak perspective approximation.

Estimates using Real Data Figure 2a shows an image of the initial position of the object we aim to recognise, and a second image of its final position after it has been rotated on the turntable. Several important features have been superimposed onto this second image. These include:

The basis which defines the affine coordinate frame on the object. It is shown as three white lines marked E_1, E_2, E_3. In this case the basis is orthogonal in 3D space, and should form a good basis for computing the invariants.

The track data for each detected point appears as a black line. During tracking one of the points was lost and is marked as such.

Labelled points (pt1-pt4) are the four points for which the Kalman Filter will estimate invariant affine coordinates.

The graphs of figure 2b show the Kalman filter estimates of the invariants for the data from figure 2a. These estimates are continuously updated as more track data is input into the Kalman Filter, such that the estimate of the invariants improves and the uncertainty is reduced. The graphs have an initial transient stage due to the initialisation of the filter with zeros.

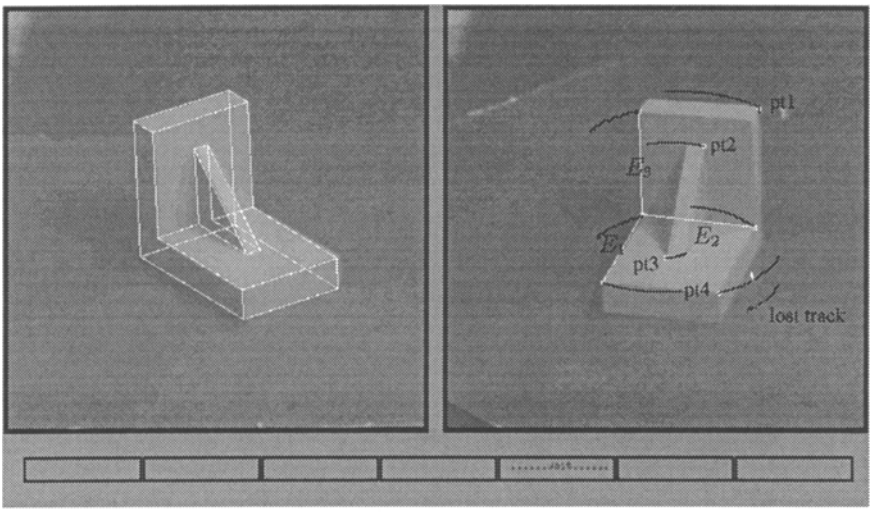
Rather than just tracking the object through an arbitrary range of viewpoints, it is preferable to *actively* be able to decide at which point to stop tracking (*i.e.* at which point the invariant estimates reach an acceptable accuracy). This can be achieved by examining the uncertainty covariance matrix and innovation returned by the Kalman Filter [12].

3.2 Geometric Hashing

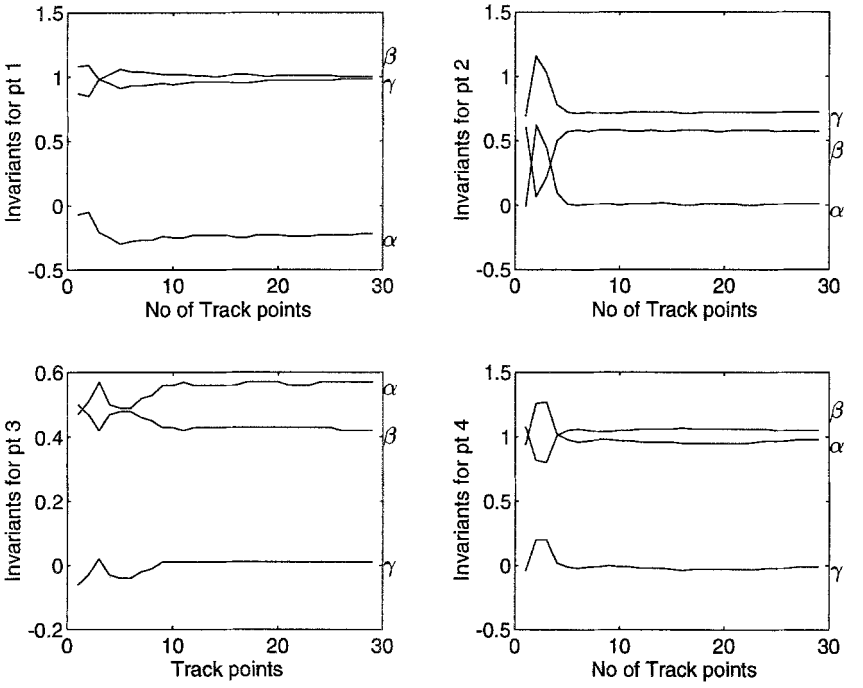
The geometric hashing approach to 2D model-based object recognition was proposed by Lamdan, Schwartz and Wolfson [13]. In this approach geometric invariants are used as a simple and stable index into a hash table. We expand on their idea and have a 3D index, $H(\alpha, \beta, \gamma)$, with each indexed location in the hash table storing information about the object used to create the invariant, the points used to form the basis and the point referenced in that basis. A large amount of pre-computation is performed on the model library to create the hash table, but since this can be done off-line it will not affect recognition speed. Invariants are calculated for all permutations of basis and points and data stored in the hash table.

An obvious problem when creating the hash table is the large quantity of data present. If an object has N_v vertices, there are $N_b = N_v!/(N_v - 4)!$ possible bases ($O(N_v^4)$), and $N_{invs} = N_b(N_v - 4)$ associated invariant sets. This problem needs to be addressed by pruning away bases, and this can be done by only selecting those bases which are well conditioned, using structural information about the object, and eliminating permutations of the same basis points. This is discussed in more detail in [12].

The effect of the uncertainty of the invariant estimates on indexing performance needs to be considered in geometric hashing, and error analysis of the uncertainty can be found in [14, 12]. The most basic information gained from this analysis is that larger invariants have larger noise components. In order to



(a)



(b)

Fig. 2. (a) Image with track data and basis superimposed. (b) Kalman Filter Outputs for 4 points in this basis. The estimate is continuously updated as more track data is input into the Kalman Filter.

compensate for this the hash table can be constructed in such a way that bins become larger for larger invariant values. The discrete nature of the bins also means some invariants lie at the limit of a bin. This problem is alleviated by searching not only the indexed bin but also its immediate neighbour. The exact implementation of this indexing is described in detail in [12].

Direct Model Acquisition from Scenes In some circumstances it may be advantageous to compile the object hash table directly from real scenes. The attraction of such an approach is that the features used in database formation and recognition are similar, since the same feature extraction process is used in both cases. Unpredictable effects such as corner detectability in the images which is unavailable in a CAD model, now form an integral part of model formation. The implementation of this scheme involves combining the results from several image sequences, in order to average out any errors in the invariants and to give an indication of likely variations. In this way complete coverage of the shape is also ensured. The approach can be viewed as having a training set for database formation, and the later recognition providing a test set. The problem with such an approach is that more human interaction is required in the form of matching points in the different sequences.

The Recognition Process Once the invariants have been estimated for a particular basis, the recognition process is completed as follows.

1. For each invariant set (α, β, γ) index into the hash table. For every (Object,Basis) grouping which appears in the associated bin register a vote in a recognition tree [12]. The recognition tree provides a fast way of accumulating votes for an (object, basis) group. In order to account for uncertainty in the 3D invariant estimate, examine not just the indexed bin but also local surrounding ones.
2. If a grouping (Object,Basis) scores a large number of votes, then this object is possibly present in the scene.
3. Compute the transformation (1) between matched model and real data points. A minimum of four points are required to calculate this transformation. At least five point matches (usually more) are available, so least squares minimisation is used to find a solution. Verify that the least squares solution to the transformation projects all model points satisfactorily to their matched image points.
4. Transform the edges of the model according to this transformation. Verify how close the projected edges lie to scene edges, if there is good correlation then the object has been recognised. A model that has been successfully projected onto the image is shown in figure 2a. Failure to match the edges requires the process to be restarted with a different basis.

Overall recognition time depends on two factors. At present the most time is spent tracking features and this will be very hardware dependent. After that estimation of invariants, indexing and backprojection is very fast but will depend on how many bases need to be tried before recognition is successful. With the simple polyhedral objects recognition is usually achieved with the first basis choice.

4 Working in Cluttered Scenes

The basic 3-dimensional object recognition system has been demonstrated to work for simple polyhedral objects, when there is an accurately localised set of corner points, from which the affine basis can be formed. Although the system can cope with some noisy points by selecting random bases until a successful one is found, this is clearly a simplistic approach which will fail in environments with many false points or where points lie on several different objects. Some information on how sets of points are interrelated is required, and *perceptual groupings* [15] provide us with a means of finding such relations.

Perceptual Groupings using Multiple Views A strong cue for grouping points which belong to the same object is connectivity between the points. However a number of problems occur with this simple inter-connectivity in cluttered scenes. Junctions of lines (where points are located in the image) are formed not only by a single object's geometry, but also when objects start occluding each other, occluding themselves or shadows start appearing (figure 4), and in a simple perceptual grouping scheme, these unwanted points can wrongly link up groups of points on different objects.

In fact these unwanted points usually correspond to T-junctions of lines which can fairly easily be detected and removed, separating the groups. Unfortunately in certain circumstances geometric junctions can align accidentally in such a way that they also resemble T-junctions. This latter type of T-junction will only be a transient effect however, and therefore important information can be gained from the *multiple views* available in our recognition system to distinguish between types of T-junctions. By tracking the *stability* of a junction through a series of views we are able to determine its significance. Junctions formed by shadows for example tend to be quite unstable and will disappear in new views, whereas T-junctions caused by occlusion will remain T-junctions. Figures 4 and 5 show examples of the same junctions seen in different views.

An *active perceptual grouping* scheme is therefore proposed based on the concept that *a useful perceptual grouping is one that remains stable through several different viewpoints.* The first image is used to initialise groupings, and subsequent views are used to update groupings.

Example Perceptual grouping based on point interconnectedness is performed on figure 4, creating a graph structure linking all the points in the scene. In order to separate the point sets belonging to the two objects, we wish to remove junction figure 4b, which is caused by occlusion and forms a T-junction. However removing all T-junctions also removes useful points like figure 4c. Information from a new view (figure 5), allows us to distinguish between types of T-junction, so we can remove junction figure 4b but not figure 4c.

The use of a single grouping cue often leads to the formation of many small disparate groups of features, because the segmentation procedure failed to locate a linking *corner* or *edge*. Additional cues must be employed: for polyhedral objects, parallelism of edges, or proximity of points can be applied. Here again

information from multiple views can be combined to give a better idea of grouping than a single image can. Figure 3 shows objects recognised using this multiple view perceptual grouping method.

5 Conclusions

The use of invariants for object recognition is clearly very appealing. They allow a single object description for many different viewpoints, and therefore provide a simple index into a shape table. A crucial requirement for any 3D object representation is that it be robust to occlusion and missing data, effects which are likely to arise in real images. Affine invariants provide such a representation since they can be evaluated for any set of just five points. More visible points simply improve the noise tolerance of the system. A common problem with many existing 3D model based object recognition systems such as Alignment [3], is that a single model is tested against the scene at a time, which can greatly reduce performance when the database contains many different objects. Geometric hashing on the other hand allows one to check against all models stored simultaneously, speeding up the recognition.

The two main problems with this approach to 3D object recognition lie firstly in the exponential growth of the hash table, a problem that has only partly been addressed [12]. It is probable that more complex features than simple corner points are required to satisfactorily solve this problem. Secondly in the difficulty of selecting four points to form a basis, which all lie on the same object and match model points. This problem is especially highlighted in cluttered scenes. Some hopeful results have been achieved using simple perceptual groupings from multiple views, but as yet it is unknown how well this will work for complex objects.

References

1. R.T. Chin and C.R. Dyer. Model-Based Recognition in Robot Vision. *ACM Computing Surveys*, 18(1):67–108, 1986.
2. D.G. Lowe. The viewpoint consistency constraint. *Int. Journal of Computer Vision*, 1:57–72, 1987.
3. D.P. Huttenlocher and S. Ullman. Recognising solid objects by alignment with an image. *Int. Journal of Computer Vision*, 5(2):195–212, 1990.
4. W.E.L. Grimson and T. Lozano-Perez. Localising Overlapping Parts by Searching the Interpretation Tree. *IEEE Trans. Pattern Analysis and Machine Intell.*, 9(4):469–482, 1987.
5. J.L. Mundy and A.Zissermann editors. *Geometric Invariance in Computer Vision.* MIT Press, 1992.
6. L.G. Roberts. Machine perception of three - dimensional solids. In J.T. Tippet, editor, *Optical and Electro-optical Information Processing.* MIT Press, 1965.
7. J.J. Koenderink and A.J. van Doorn. Affine structure from motion. *J. Opt. Soc. America*, 8(2):377–385, 1991.
8. D. Weinshall. Model-Based Invariants for 3-D Vision. *Int. Journal of Computer Vision*, 10(1):27–42, 1993.

9. J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):51–68, January 1993.
10. R. Fawcett, A. Zisserman, and M. Brady. Extracting structure from an affine view of a 3D point set with one or two bilateral symmetries. In *Proceedings of the British Machine Vision Conference*, pages 349–358, Guildford, 1993.
11. S. Vinther and R. Cipolla. Towards 3D object model acquisition and recognition using 3D affine invariants. In *Proc. British Machine Vision Conference 1993*, pages 369–378, 1993.
12. S. Vinther and R. Cipolla. Active 3D object recognition using 3D affine invariants. Technical Report CUED / F - INFENG / TR164, Dept. of Engineering, University of Cambridge, 1994.
13. Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Affine Invariant Model-Based Object Recognition. *IEEE Trans. on Robotics and Automation*, 6(5):578–589, 1990.
14. W.E. Grimson, D.P. Huttenlocher, and D. Jacobs. A study of affine matching with bounded sensor error. In *Proc. 2rd European Conf. on Computer Vision*, 1992.
15. D.G. Lowe. Three dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

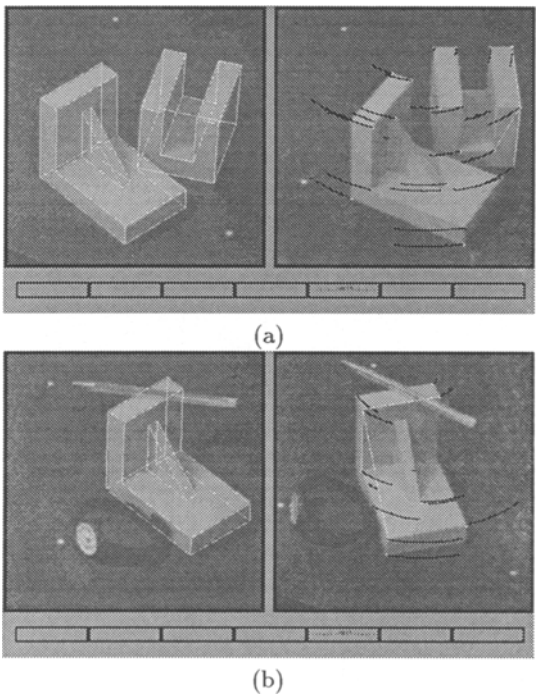


Fig. 3. a) Multiple objects recognised in a scene. b) Object recognised in a cluttered scene. T-junctions are removed using multiple view information.

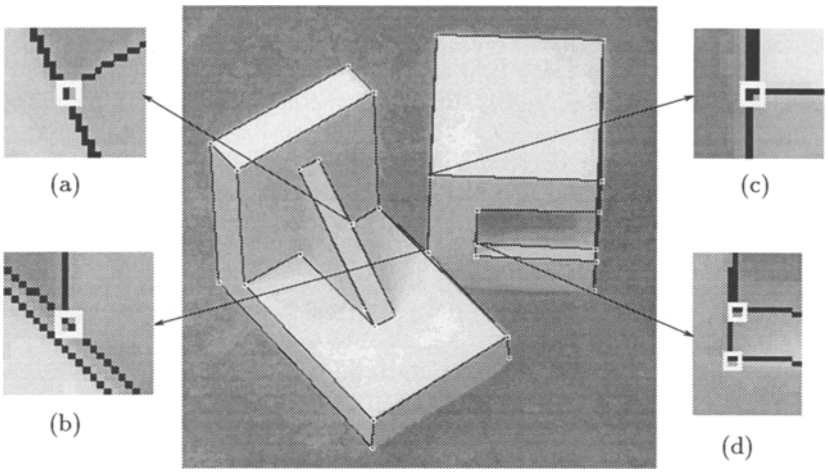


Fig. 4. A Variety of T-junctions caused by: (a) self-occlusion (b) occlusion (c) accidental alignment (d) shadows

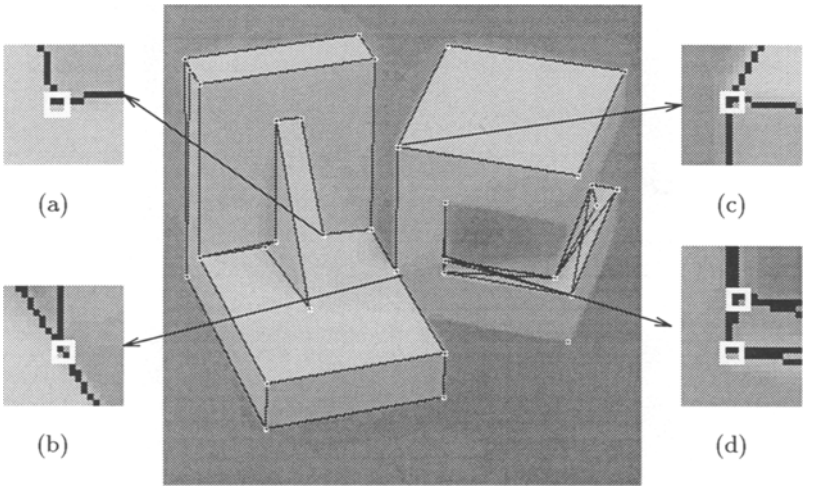


Fig. 5. The T-junctions as seen from a different viewpoint. In particular (b) has remained a stable T-junction, making it likely it was caused by occlusion. (c) which was caused by accidental alignment is no longer a T-junction. By tracking the *stability* of a junction through a whole series of views we are able to determine how it affects the perceptual groupings.