

Stereo-matching of line-segments based on a 3-dimensional heuristic with potential for parallel implementation.

R.L. Vergnet, S.B. Pollard and J.E.W. Mayhew.

Artificial Intelligence Vision Research Unit,
University of Sheffield, Sheffield, S10 2TN.

This work provides a segment-based alternative to the edge-based stereo algorithm already existing in the TINA system. Our starting point is the algorithm of Ayache & Faverjon (1985). It attempts to recover groups of line-matches having low local disparity variations. In our implementation of Ayache & Faverjon's algorithm matches were built quickly, but glaring mistakes occurred and the general behaviour was difficult to control. Therefore the constraints on the continuity of the world implicitly used in the original approach were reformulated to achieve more robust matching, more particularly by requiring mutual support between reconstructed 3-D segments. A new algorithm has been designed that is suitable for parallel implementation, where 3-D matches and their neighbourhood relationships are explicitly computed, cliques found (objects) and uniqueness enforced.

This paper describes a line based approach to binocular stereo that augments the capabilities of the TINA system (Porrill et al 1987, Pollard et al 1987). The already existing stereo matching is edge-based utilising the PMF algorithm (Pollard 1985) and its descendents. Whilst an edge based approach is more general, image structures that are amenable to description by linear approximations can be efficiently and robustly matched by an alternative strategy. Hence a combined approach is adopted allowing either, redundant *cross* primitive consistency to give increased robustness, or segment matching to *seed* matching at the lower level to improve efficiency.

The starting point for this work is the segment matching algorithm due to Ayache and Faverjon (1985) which employs a hypothesis, propagation and test strategy to identify corresponding segments. Only a small number of well selected segments act as a basis for match hypotheses, the remainder being found as a result of the propagation stage. Their algorithm sets up a few likely correct matches and propagates them. Along the propagation, a weak disparity gradient constraint is enforced through an imposed limit upon the disparity difference between neighbouring matched segments.

R.L. Vergnet is a visiting research student from ENST, 46 rue Barrault, 75013 Paris, and gratefully acknowledges the British Council for their support.

In this way the algorithm can be thought to be identifying the most geometrically plausible three-dimensional structures from the pool of potential matches by building them recursively. Once a few but strong hypotheses have been propagated independently, the final selection of matches is achieved preferring those generated by the most successfully propagated hypotheses.

One disadvantage of the Ayache and Faverjon's algorithm is its unsuitability for parallel implementation. The finest grain of potential parallel computation is the propagation of the match hypothesis. In our serial implementation of their algorithm only a small number of hypotheses (between 5 and 15) are usually necessary to achieve full matching, and only the correct hypotheses (a small proportion) account for the largest proportion of computational effort because they propagate most widely. Furthermore, there is great redundancy in the matches raised by these good hypotheses. The improvement of performance with increasing number of processors soon achieves saturation.

This paper describes an alternative segment based algorithm more suited to a parallel approach. While our current implementation is on a sequential machine it is our intention to develop the algorithm to run on MARVIN the AIVRU transputer-based vision architecture.

The other major departure from Ayache and Faverjon's original algorithm is that we prefer an implementation of the continuity constraint based upon actual three dimensional connectivity rather than the computationally less expensive alternative of rough disparity similarity. The more powerful constraint actually involves little extra computational overhead when a simplified parallel representation of the camera geometry is used.

1. LINE DESCRIPTION

1.1. Geometry and Representation

Edges are obtained from Canny operator (1986) at a single high frequency (small sigma Gaussian smoothing). Following their detection, edge-strings are formed by linking edge pixels using a few simple heuristics. Straight-line approximation employs a recursive fit (by orthogonal regression) and segment strategy. Segmentation points are added when the edge string line deviates from the current line fit. Robustness and efficiency are obtained through the use of a heuristic search strategy to identify those regions of strings/sub-strings that are most amenable to

straight-line fit.

We have found this scheme to be more reliable at *identifying* true straight lines consistently than more traditional polygonal approximations usually intended to segment edge-strings for data-compression purposes. However this strategy proves to be less able to deal with data that does not arise from linear features.

Line segments are rectified; that is they are reprojected into a parallel camera geometry equivalent to the original non-parallel one. This has the effect that the epipolar constraint becomes a same raster constraint and the process of 3-D projection is greatly simplified.

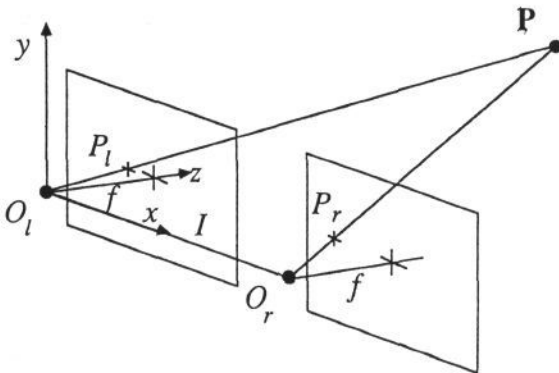
The left and right cameras have optic centers O_l and O_r , respectively. The principal axes are perpendicular to the interocular axis that connects the optic centers. Coplanar image planes, parallel to the interocular axis are illustrated in front of the optic centers for simplicity. The world coordinate frame is located at the optic centre of the left camera with axes along the focal direction (z : positive towards the scene), the interocular direction (x : positive away from the left optic center), and the direction mutually orthogonal to these (y : positive upwards). The coordinate frames of each image are located where the principal axes intersect their respective image planes with the x and y axes aligned with those of the world frame.

Consider a point P at (x, y, z) and its projection into the left and right images at $P_l = (X_l, Y)$ and $P_r = (X_r, Y)$ respectively. The disparity d between these projections is given by $d = X_l - X_r$.

It follows from similar triangles and algebraic manipulation that

$$x = \frac{X_l I}{d} \quad y = \frac{Y I}{d} \quad z = \frac{f I}{d}$$

where f is the focal length of the cameras, and I the distance between their optic centres.



Line segments are denoted L_l and L_r in the left and right images respectively and as L in the 3-D world. Each line segment is represented in the overdetermined fashion by the attribute tuple

$$(P_1, P_2, V, C, con, length)$$

Where:

- P_1 and P_2 are end point vectors
- V is the direction P_1 to P_2
- C the centroid
- con the average measured contrast along the line (or its projection into the left image) which is measured in the direction V with positive polarity defined as "dark to the left/light to the right"
- $length$ is the length of the line

1.2. Windows

Each image has a window bucket structure over the line structure to provide efficient spatial access. Each image is segmented into 16 by 16 grid of windows. Each window has a line "bucket", i.e a list of references to the lines that intersect the window (these are accessed bucket[i][j] for the window row i col j).

Additionally each line has a list of windows it intersects. Window buckets simplify the search for match candidates. Search is restricted to the set of window buckets that lie within the disparity search interval (which is of limited extent). As shown below, buckets are also used for fast computation of image-based segment neighbourhood relationships

1.3. Line Matching

Line matches are identified provided they satisfy the following criteria (all these criteria are verified symmetrically on a left-to-right *and* right-to-left basis):

- (1) Only line features for which
 - $length \geq l_{min}$.
 - $V \cdot i \geq \cos(\theta_{min})$ where i is the horizontal axis direction.
 are used as matching primitives.

Small edge-strings are very often noisy and seldom describe straight features of the scene. Therefore a thresholding is made on the length of the segments. θ_{min} is an orientation threshold in order to eliminate horizontal segments for which 3-D computation is difficult.

- (2)
$$\frac{|V_x/V_y - V_r/V_r_y|}{\sqrt{(V_x/V_y)^2 + (V_r/V_r_y)^2 + 1}} \leq K$$
 : a disparity gradient constraint for orientation.

We prefer this orientation constraint because the disparity gradient is related to the slope of the 3-D lines of the scene with respect to the z -axis. The definition and use of the disparity gradient limit can be found in (Pollard PMF 1985). With this formulation, the limit on reorientation is tuned according to the orientation itself. Thus, the angular difference is maximal for vertical segments ($V_x = 0$) and null for horizontal ones ($V_y = 0$). The value of K is set between 1 and 3.

$$(3) \quad \frac{1}{c} \leq \frac{con_l}{con_r} \leq c : \text{a similar contrast constraint.}$$

Since the contrast criterion is not very robust, the value of c is set relatively high (typically 3, 4). However the mere sign of the contrast is still a strongly discriminating feature.

- (4) Matches for L_l must be intersected by the raster of the centroid $C_l = (X_l, Y_l)$ of L_l .

This is the epipolar constraint suitably adapted for whole segments.

- (5) Matches must lie within the disparity range $[D_{low}, D_{high}]$; that is matches for L_l must have X_r in the range $[X_l - D_{high}, X_l - D_{low}]$.

The limit on the allowed range for disparity is derived from the possible positions of the features of the scene and corresponds to a depth range $[z_{min}, z_{max}]$.

Given the existence of a match between L_l and L_r we compute L as follows:

- (i) First identify the intersection of L_l and L_r in the vertical direction, call this Y_{low} to Y_{high} and the points of intersection of the of L_l and L_r with these "rasters" $P_{l_{low}}, P_{l_{high}}, P_{r_{low}},$ and $P_{r_{high}}$.
- (ii) The 3-D end points of L are computed from matches $P_{l_{low}} \rightarrow P_{r_{low}}$, and $P_{l_{high}} \rightarrow P_{r_{high}}$ respectively to give P_{low} and P_{high} .
- (iii) The other attributes (except contrast) of L are computed from P_{low} and P_{high} .

Thus a set of 3-D lines involving a same left line L_l is generated:

$$L_l \rightarrow \{ L_l^1, L_l^2, \dots, L_l^{n_l} \}$$

Note that it is the intersection of the left and right image features that give rise to the 3-D segment. A less conservative approach would have been to adopt either left, or right image feature (or their union) to provided the vertical extent of the image projection of the 3-D primitive.

2. ALGORITHM OVERVIEW

Our basic approach is to construct *all possible* line match structures (in 3-D) prior to a stage of parallel constraint propagation. The limiting resolution of realistic parallel implementation being the line segment. In order to simplify the structure of the algorithm verification of matches is performed with respect to one image only (the left). Unlike Ayache and Faverjon we allow line segments to have multiple matches provided they satisfy certain geometrical constraints and do not intersect common rasters.

The algorithm employs the following heuristics to exploit the assumption of locally continuous 3-D space:

- (1) We define a 3-D line neighbourhood, saying that two lines L_1, L_2 are neighbours when their distance $D(L_1, L_2)$ is less than a certain threshold ϵ . The distance measure is not the minimum Euclidian distance between L_1 and L_2 but the minimum end point distance. A continuous object is defined as a set of lines that are directly or indirectly linked by neighbourhood pairwise relationships. Continuous objects that contain greater numbers of constituents are preferred.
- (2) Within objects, a line-segment often will have several immediate neighbours therefore line interpretations that possess greater numbers of 3-D neighbours will be preferred.

In order to restrict the 3-D neighbourhood search, A 2-D image based neighbourhood graph is computed. To each left line L_i a list of its neighbours is attached:

$$L_i \rightarrow \{ L_i^1, L_i^2, \dots, L_i^{n_i} \}$$

If the distance between 3-D lines is less than ϵ , then their projections in the images are expected to be less than a certain limit

$$v = \frac{f\epsilon}{z_{min}}$$

where z_{min} is a chosen minimum depth.

In the two images neighbours are computed using the buckets, as follows :

- For each line L , look at the set of buckets attached to L (ie that it intersects).
- Given subscripts i and j of a window corresponding to one such bucket, consider the eight neighbours of that bucket that differ in subscript at most one.
- Among the lines contained in these buckets, select the lines L' such that $dist(L, L') \leq v$.

3. VALIDATION OF THE MATCHES

When all the possible matches have been made and the corresponding 3-D segments computed, for each proposed 3-D segment L the following attributes are maintained:

- *match* L_r
- *3-D_neighbs*
- *object*
- *obsz*
- *nlinks*

L_r is the right line that is matched to L_l . The idea is to enforce uniqueness constraint on the matches by eliminating *ghost* (incorrect) 3-D lines on the basis of the continuity of the 3-D world. Within continuous 3-D objects, we give more support to the lines that have a great number of neighbours.

The algorithm proceeds as follows (see schematic figure in end of this section):

(i) Compute 3-D neighbourhood graph:

A pair of lines L and L' are considered to be neighbours in 3-D if:

L_l and L'_l are neighbours : where L_l and L'_l are the left image line primitives involved in the generation of L and L' respectively.

$$dist (L, L') \leq \epsilon$$

Furthermore a pair of lines L and L' that have the same left image line primitive L_l are considered to be neighbours if:

L_r and L'_r are collinear and non overlapping: where L_r and L'_r are the right image line primitives involved in the generation of L and L' respectively.

$$dist (L, L') \leq \epsilon$$

(ii) Compute connected sets:

Propagation is performed along the links of the 3-D neighbourhood graph to identify and label connected sub-graphs. The attributes of each line are set as follows:

- $nlinks$ = number of other segments L' it is connected to directly.
- $object$ = number of the object/sub-graph the line belongs to.
- $obsize$ = size of the object/sub-graph the line belongs to ie the number of lines forming the object.

Note that each object is not necessarily consistent. It is possible (highly likely) that more than one 3-D line segment resulting from matches of the same left or right image primitive are included in the same object, most of them being incorrect. In this way it is possible for "leakage" between objects to occur (because of *ghost* segments that result their merger to a single entity).

(iii) Enforce *within-object* uniqueness:

Ideally within object uniqueness would be computed using some form of "max-clique" algorithm subject to the constraint that matches are consistent except if they involve the same left or right image primitive. The maximal consistent clique found would provide a definition for a potential object.

The strategy we prefer on grounds of computational efficiency and potential for parallel implementation is to remove from each object/sub-graph all except the most strongly supported amongst any set of lines that violate the uniqueness constraint with respect to each other. Support is determined directly in terms of $nlinks$ score ; the immediate neighbourhood support. Uniqueness is defined over lines L_l from the left image and allows only single matches except where they could arise from a single 3-D line feature in which case they are required to be collinear the right image.

Note that within-object uniqueness (and uniqueness in general) is determined with respect to the left image alone. It would be straight forward to extend the definition of uniqueness to include the right image, however we have found the simpler approach to be sufficient for our purposes.

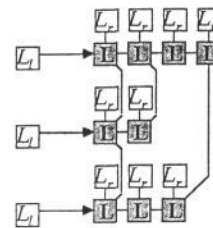
- (iv) Stage (ii) is repeated to identify new connected objects that are now consistent with the uniqueness constraint (but not necessarily optimal). Discarded segments are not allowed to form part of these objects. Accordingly it is possible that fragmentation of the former objects into sub-objects will occur at this point.

To summarise: on the first pass, objects are computed, and then segmented from low-support lines. On the subsequent pass smaller objects are identified that satisfy within-object uniqueness.

- (v) Enforce *global* uniqueness : for each L_l and for each pair (L, L') such that they belong to a different objects, choose the one that belong to the biggest object, ie that has the highest *obsize*, and eliminate the other.

- (vi) Remove the objects that have a *obsize* below threshold for it is very probable that they are ghost objects.

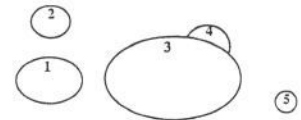
LINE MATCHES



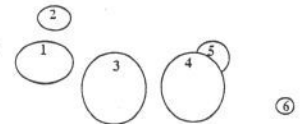
The prediction of hypotheses gives raise to matches between left and right lines, from which 3D segments L are computed. Then 3D neighbourhood relations between lines L are set.

VALIDATION OF THE MATCHES

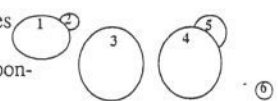
- I. Connected sets of neighbouring segments are found , labelled and their size computed.



- II. Within object uniqueness is enforced on the basis of local support, therefore lines are removed and objects are recomputed (3 has split into 3 and 4).



- III. Global uniqueness with respect to left lines is enforced on the basis of object size. (2 has been cleared from lines corresponding to left 2D lines involved in 1,3 ...)



- IV. Objects whose size is too small are removed for being supposedly ghost objects.



4. EXAMPLES OF PERFORMANCE

Examples of performance on two stereo pairs are shown in the last figures. It must be pointed out that even with loose constraints on local features, the way from ambiguous to disambiguated stereo-data is not very long.

On Fig a & b, we show pairs of polygonal approximations, then 3 stages of the algorithm are shown, and on a purpose of intelligibility, with two different 3-D views of the same scene for each step.

(a₁, b₁) All the matches (two views).

(a₂, b₂) After within-object uniqueness (two views).

(a₃, b₃) After global uniqueness (two views).

The parameters are set to the following values :

MATCHING

- $length > 4$ pix (for the Boxes 512x512 and 12 pix for the House 256x256)

- $\theta_{min} = 8$ deg

- $K = 1.5, c = 4$

VALIDATION

- $\epsilon = 6$ mm, $obsize \geq 3$

5. CONCLUSION

As the algorithm is data-oriented, it can be implemented in parallel without having to be redesigned. The data can be split so that it is shared and processed separately, provided fast communication channels between processors are available.

Enforcing the uniqueness constraint rigorously allows the algorithm to rely only loosely on image-to-image similarity criteria and therefore achieve a more robust matching. Note that there is no constraint on the length of the segments.

Working with 3-D primitives allows us to exploit powerful heuristics based directly upon 3-D constraints. It is well known that horizontal lines present fundamental ambiguity to binocular stereo algorithms in general. However, in the algorithm above it is possible by reasoning over the polygonal scene structure to hypothesise and verify the 3-D position of horizontal lines.

Finally, a few weaknesses (or at least characteristics) of our approach must be underlined :

- The value of the neighbouring threshold ϵ is scaled on the scene (is it a drawback?).
- There is a limitation to scenes amenable to canonical representation in terms of straight-line image primitives.

REFERENCES

N. Ayache "Construction et Fusion de Représentations Visuelles" Thèse de Doctorat ès Sciences, Univ. de Paris-Sud, Mai 1988.

N. Ayache and **B. Faverjon** "Fast Stereo Matching of Edge Segments Using Prediction and Verification of Hypotheses" INRIA (France) Short Paper 1985.

J. Canny "A computational approach to edge detection" *IEEE Trans. Pattern Anal. and Machine Intell.* Vol PAMI-8 pp 679-698

R. Horaud, T. Skordas "Stereo Correspondence Through Feature Grouping and Maximal Cliques" LIFIA-IMAG, D-LETI (Grenoble) to appear in *IEEE Trans. Pattern Anal. and Machine Intell.*

G. Medioni and R. Nevatia "Matching Images Using Linear Features" *IEEE Trans. Pattern Anal. and Machine Intell.* Vol PAMI-6, No 6, November 1984.

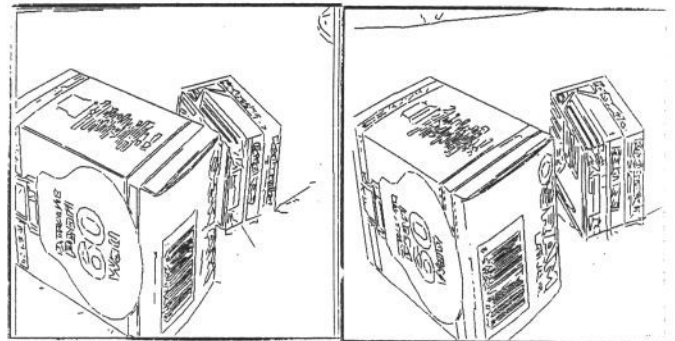
G. Medioni and R. Nevatia "Segment-Based Stereo Matching" *Comp. Vision, Graph. and Image Proc.* Vol 31, pp 2-18 (1985)

S.B. Pollard "A stereo Correspondence Algorithm using a disparity gradient limit", *Perception* Vol 14, 1985, 449-470.

S B Pollard, T P Pridmore, J Porrill, J E W Mayhew, and J P Frisby "Components of a Vision Stereo System" *Proc. Fourth International Symposium of Robotics Research, MIT Press, pp 229-235.*

S B Pollard, T P Pridmore, J B Bowen, J E W Mayhew, and J P Frisby "TINA : The Sheffield Vision System" *Proc. Ninth International Joint Conference on Artificial Intelligence, Milan, pp 1138-1144.*

Fig a



Segment-based Stereo-pairs

Fig b

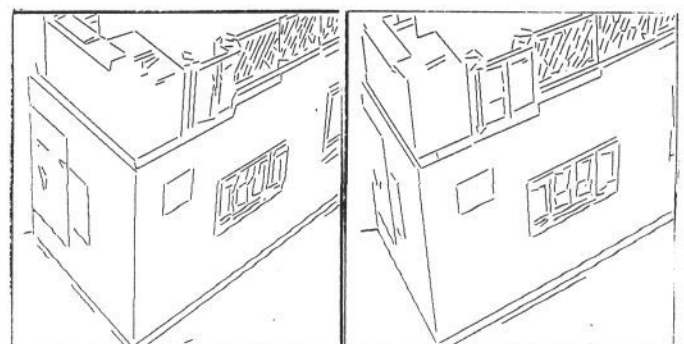


Fig a₁

All matches

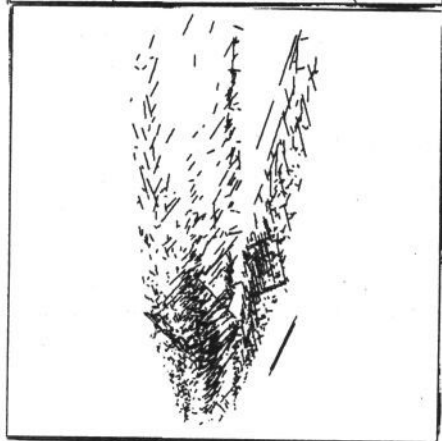
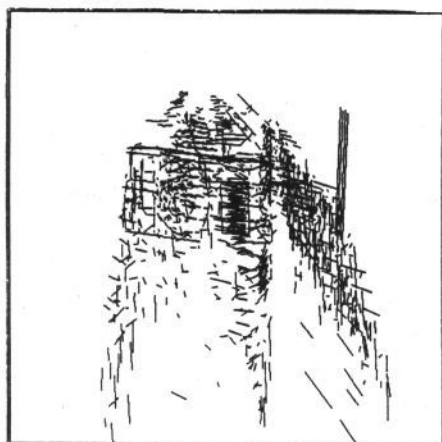


Fig a₂

Within-object uniqueness

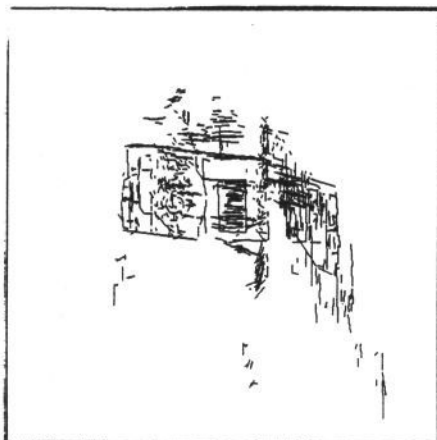


Fig a₃

Final matches

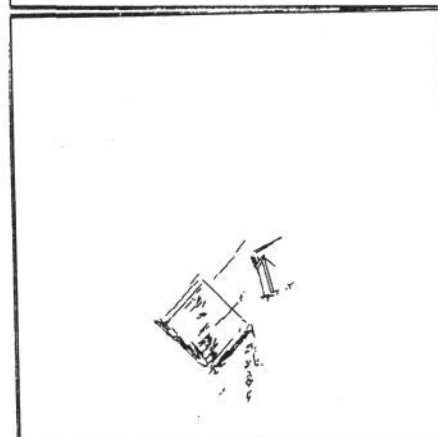
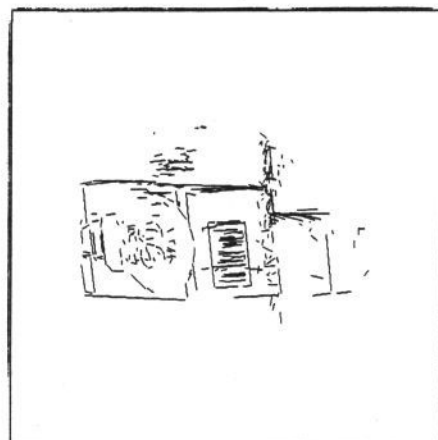


Fig b₁

All matches

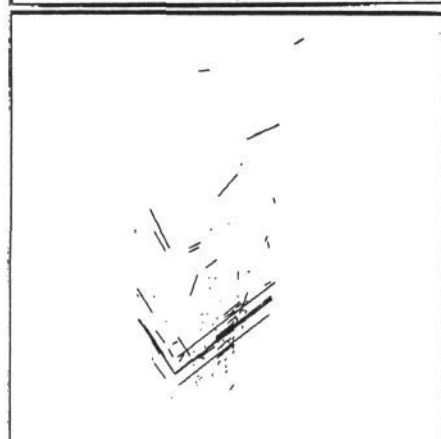


Fig b₂

Within-object uniqueness

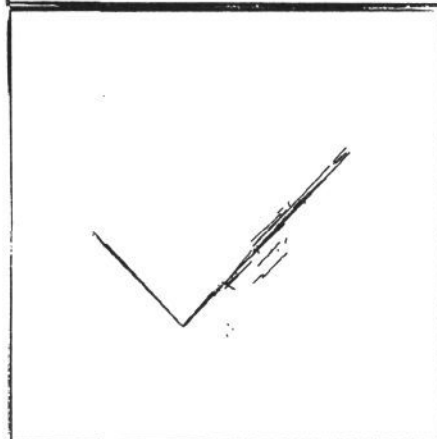
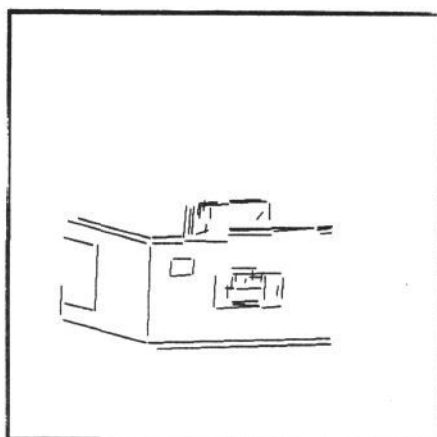


Fig b₃

Final matches

