Stereo Correspondence Through Feature Grouping and Maximal Cliques

RADU HORAUD AND THOMAS SKORDAS

Abstract—In this paper we propose a method to solve the stereo correspondence problem. The method matches features and feature relationships and can be paraphrased as follows. Linear edge segments are extracted from both the left and right images. Each such segment is characterized by its position and orientation in the image as well as its relationships with the nearby segments. A *relational graph* is thus built from each image. For each segment in one image a set of potential assignments in the other image is determined. These assignments are represented as nodes in a *correspondence graph*. Arcs in this graph represent compatible assignments established on the basis of segment relationships. Stereo matching becomes equivalent to searching for sets of mutually compatible nodes in this graph. These sets are found by looking for maximal cliques. The maximal clique the best suited to represent a stereo correspondence is selected using a benefit function. Finally we show numerous results obtained with this method.

Index Terms—Feature-based matching, feature grouping, geometric constraints, maximal cliques, stereo vision, subgraph isomorphism.

I. Introduction

THE stereo correspondence problem is the problem of matching two images of the same scene from different viewing positions. As a consequence, the three-dimensional geometry of the scene may be recovered. The correspondence problem is difficult because it is not straightforward to find a one-to-one mapping between the two images. A space point projects onto the two images at two different locations and in the absence of additional knowledge it is practically impossible to establish a relationship between these locations. In this paper we propose a solution to the correspondence problem based on the exploration of both the geometric constraints available with the imaging device and the structural similarities between the two images. On one hand, the known relationship between the two images imposes metric constraints on the relationship between the two projections of a space feature. On the other hand, certain descriptive properties of the scene are quasi invariant under perspective projection, and therefore they do not change too much with the view-

ing position. We propose and implement a matching algorithm which explicitly takes into account both these pieces of information: metric constraints and structural (descriptive) similarities between the left and right images. This algorithm constructs a graph and then performs a graph search in order to find the best available sets of mutually compatible left-to-right assignments.

A. Approaches to Stereo

Previous approaches to the stereo correspondence problem have attempted to take into account various pieces of knowledge in order to simplify the search. See [6] for a discussion concerning research in stereo vision performed before 1982. More recently, image features have extensively been used as a basis for stereo matching. We briefly review the constraints that are used in conjunction with image features in order to solve for the stereo correspondence problem.

The dimensionality of the search space can be reduced from two dimensions to one dimension by observing that for any point in one image, its potential matching points in the other image must lie along a line: this is the *epipolar constraint*, a geometric property available with the stereo sensor whenever it is properly calibrated. Furthermore, opaque surfaces impose an *ordering constraint* along corresponding epipolar lines. The ordering constraint has been used in various ways, i.e., [4], [17] and has been thoroughly studied in [22]. The gradient of the *disparity* (the disparity being defined as the difference in position between the two images of a projected space point) is directly related to the smoothness of the scene surfaces. If one deals with smooth surfaces, the disparity will vary smoothly. The smoothness of disparity constraint is used in most of the existing techniques. Edge-based matching making use of multiple scale representation is proposed by Grimson [13]. Eastman and Waxman, [11] suggest to include correspondence and reconstruction into a unique stage using an explicit surface representation. A limit on the disparity gradient is used in [19] in order to eliminate impossible pairs of matches.

All these approaches usually assume that the matching features are markings on smooth surfaces. The proposed algorithms work well especially in such domains as aerial photography interpretation and artificially made surfaces (random dot stereograms, for example).

While it seems to be correct to use the smoothness of disparity constraint for matching surface markings such

as texture edges, it is not correct to apply this constraint when dealing with discontinuity edges, i.e., edges arising from surface orientation discontinuities or from depth discontinuities. Indeed, at these locations the scene surface is not smooth and hence the smoothness of disparity assumption is not valid any more. As a complimentary approach we want to investigate ways to perform stereo matching without using this constraint. Instead, we want to explore image feature relationships.

A. Our Approach

In this paper we propose a method for matching straight lines and relationships between them. Hence, we emphasize the cooperation between feature grouping and stereo matching.

Straight lines are extracted from each image. These lines are grouped on the premise that some scene properties are invariant under perspective projection. Hence, an image is described in terms of features (straight lines), feature attributes, and relationships between nearby features. The *structural description* thus obtained is represented by a relational graph. With this representation an image is considered globally rather than as a list of individual features.

The stereo correspondence problem becomes the problem of finding a match between the two structural descriptions, i.e., a mapping function between elements of the two sets of features which preserves the compatibilities between feature relations. We denote by l_1 and l_2 two left image features and by r_1 and r_2 two right image features. Let $\Re_1$ be the relation between l_1 and l_2 and $\Re_2$ be the relation between r_1 and r_2. The mapping function ($l_1 \rightarrow r_1$), ($l_2 \rightarrow r_2$) must satisfy the following conditions.

1) The relation $\Re_1$ between l_1 and l_2 must be *compatible* with the relation $\Re_2$ between r_1 and r_2.

2) The mapping is one-to-one,[1] i.e., each feature in the left image is assigned a single feature in the right image.

3) The mapping must maximize a benefit function.

In order to find such a mapping function we propose to build a correspondence graph and to perform a search in this graph. For each feature in one image we compute the range of expected geometric characteristics for its potential assignments in the other image. A certain number of left-feature-to-right-feature pairs is obtained. These pairs constitute the nodes of the correspondence graph. Whenever two such pairs are compatible, an arc is built between the corresponding nodes. A stereo matching, or a mapping function is equivalent to a set of mutually compatible nodes. Hence, the stereo correspondence problem is now cast into the problem of finding maximal cliques in a graph.

The largest maximal clique in the correspondence graph is associated with the mapping which has the largest number of feature pairings and the largest number of compatible relations. Hence, the largest maximal clique should

be regarded as the "best match" [5]. In practice we have noticed that among the many maximal cliques detected in the correspondence graph, the largest one does not always correspond to the best match. Moreover, there may be several largest maximal cliques. Therefore we have associated a benefit function to each maximal clique. This benefit function advantages "large" cliques. The maximal clique which maximizes this benefit function (not necessarily among the largest maximal cliques) is said to be the best available correspondence.

One fundamental contribution of our approach is a technique which allows the construction of the correspondence graph from two relational graphs which, in the particular context of stereo vision are quite different. This discrepancy reflects imperfect similarities between the two images due to occlusions, accidental alignment, feature inversion (local violation of the ordering constraint), and/ or failures of the feature extraction process.

Our approach is best shown on the following figures. Fig. 6 shows two relational graphs obtained from the images of Fig. 5 and which have to be matched. Our algorithm found 9 solutions corresponding to 9 maximal cliques in the correspondence graph. These solutions are shown in Fig. 9. In this case the best solution corresponds to the unique largest maximal clique. This solution is shown in Fig. 9(a).

C. Related Work

Linear segment matching in conjunction with graph search is used both by Medioni and Nevatia [16], and by Ayache and Faverjon [3]. However, these approaches have considered image features individually. Relationships between these features have been used only weakly. Ayache and Faverjon propose a method which first selects a small set of left-to-right matches and second attempts to grow this match to include nearby features[2] with the heuristic that for neighboring image features the disparity varies only slowly. This method is implemented as a depth-first tree search associated with a hypothesize-and-test strategy. The method is elegant because it avoids exhaustive search but there is no guarantee that it finds neither the largest nor the best set of available matches. The method of Ayache and Faverjon matches straight lines. Our method matches straight lines and relationships between them. These relationships are useful both for stereo (as advocated in this paper) and for constructing a symbolic scene description and hence they need to be extracted anyway.

Lim and Binford [15] and Herman and Kanade [14] perform junction matching. The space of potential matches contains a set of nodes where each node represents a left-junction-to-right-junction assignment. In [14] each node has a cost associated with it. To arrive to a unique set of junction matches, the set of potential matches is searched for a minimum-cost path. The cost of a path is the sum of

---

[1] In fact, an exception to this uniqueness condition is allowed as it is explained in Section IV-B.

[2] In Ayache and Faverjon's work, nearby features means features that are close to each other.

its nodes' costs. The implemented algorithm does not guarantee to result in the lowest cost path.

Moreover, the stereo matching method proposed in [14] takes into account geometric knowledge available with the processed scenes. For example, the fact that in urban scenes the buildings' roofs tend to be parallel to the ground plane, while walls tend to be perpendicular to this plane. Our stereo matcher is more general since it does not make use of this kind of domain-dependent knowledge.

These stereo techniques do not explore constraints that allow to assert mutual compatibilities between matches. With respect to these approaches we use a larger set of feature relationships and we build a correspondence graph in which a node represents a left-line-to-right-line assignment; the belonging of a line to a feature group is used to explicitly establish direct compatibility (or incompatibility) between nodes.

Recently, a structural stereo matching method was described by Boyer and Kak [9]. They propose a cost function derived from information theory which they use in conjunction with heuristic tree search. They obtain good results with matching skeletal primitives extracted from elongated objects. Although they define structural matching mathematically they never make explicit what they really mean by structure from a computer vision viewpoint and it is not clear what structural properties are to be used.

In the past, feature grouping and maximal cliques have been used for matching a geometric model to either 2-D data [1], [7] or 3-D data [8]. We appear to be the first to apply maximal-clique graph search to stereo matching. However, we do not have the luxury of a perfect geometric model. In our case, no rigidity assumption is made on the relational structures to be matched as is the case with the object recognition methods mentioned above.

D. Paper Organization

The remainder of this paper is organized as follows. Section II describes the monocular analysis which is applied simultaneously to both images. Section III describes the geometry of the stereo sensor and the constraints associated with it. Section IV describes a graph representation as well as the construction of this graph that is well suited for stereo matching. Section V describes an implementation of the proposed algorithm and shows some experimental results obtained with indoor and outdoor scenes. Finally, Section VI discusses the main advantages and limitations of the proposed method and indicates directions for future work.

II. FEATURE EXTRACTION, GROUPING, AND MONOCULAR DESCRIPTION

This section presents the monocular structural description used by the stereo matcher, as well as how this description is extracted from the raw intensity data.

An image array contains two pieces of information: Light intensity changes and local geometry. A *rich* description must incorporate explicit representations of both these pieces of information. Feature extraction is a key process for many vision algorithms. Surprisingly enough, feature grouping has somehow been neglected. In particular, as it was outlined in the previous section, only few stereo matching techniques use grouping. We believe that feature grouping is essential because it reduces the combinatorial explosion of the search space associated with the stereo correspondence problem.

It has long been argued that stereo correspondence algorithms should not rely on the presence of monocular cues. However, in practice any image, including random dots, does contain some kind of features. One has to select the most appropriate features for matching. On one hand, very low-level features such as isolated edges increase the complexity of the search process. On the other hand, high-level features produce only a sparse depth map and limit the domain of applicability of the method to those scenes containing the selected features. Straight lines seem to be a good compromise and hence they have been selected as candidates for matching. Moreover, straight lines naturally incorporate the figural continuity constraint.

Our analysis starts with a classical line detection process: Edge detection, edge linking [10], and piecewise segmentation [18]. This analysis is quickly expanded to include the *adjoining regions* and the *connecting lines*. The process of extracting these local image configurations is referred to as feature grouping and is described in detail in [20].

Each line has to adjoining regions, one on its left and one on its right. Such an adjoining region is a *stripe* along a straight line. The way such a stripe is detected guarantees that there is no edge (and hence no line) inside it. The detection of these stripes allows us:

1) To compute a contrast associated with each line. This is a weak property (just like any other raw photometric feature) especially with respect to the presence of occlusions which may occur in the scene. Nevertheless, if properly combined with other constraints it allows the elimination of very unlikely match pairs;

2) To orient the line up to 360°; and the most important

3) To establish a list of lines immediately on the left side of the current line and another list of lines immediately on its right side.

These stripes are detected as follows. Consider a point belonging to the line under analysis. From this point, a pixel-by-pixel displacement is performed in a direction perpendicular to the line. This is done to the left and to the right, until an edge point is encountered. This procedure is repeated for every line point. Each stripe thus detected is bounded by the line itself on one side and by a set of edges on the opposite side. Each stripe is also characterized by the mean value of its pixels' gray-levels. The line can now be oriented relatively to these stripes such that the darker stripe lies on its left side and the brighter stripe lies on its right side. This local structure is best shown in Fig. 1.
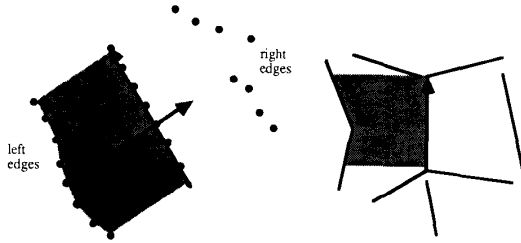
Fig. 1. A line and the detection of its adjoining regions (left), and the final local configuration associated with a line (right).
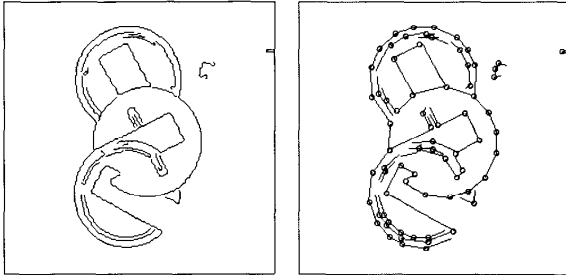

Fig. 2. Edges extracted from curved objects (left) and the associated structural description (right). The *left_of* and *right_of* relations are not shown.

Let us consider now the two sets of edge points just defined, i.e., bounding the left and right stripes, respectively. They may belong to lines previously found by the line detection process. Consequently, each line has a left stripe and associated with it, a set of left lines. Similarly, it has a right stripe and associated with it, a set of right lines. Notice however that one of these sets may be empty because the line is close to the image border.

Each line has two sets of connecting lines (two junctions) which are associated with each one of its ends. See Fig. 1. A junction is defined as a set of at least two lines passing through a common point. Junction detection is thoroughly described elsewhere, [20]. Within a junction, two collinear lines play an important role. Collinearity is invariant under projection and hence it will not vary with the changing viewing position: It is therefore an interesting property to be used in stereo matching.

In conclusion, the set of image lines may be represented by a set of nodes, where each node represents a line with its properties-position, orientation, length, and contrast,[3] and by a network of pointers, where each pointer represents a relation between two nearby lines. These relations are *left_of*, *right_of*, *collinear_with*, and *same_junction_as*. It is worthwhile to notice that these relations are symmetric. We call this network a *monocular* description. This description partially captures the image structure. As an example, Fig. 5 shows two sets of lines (the arrows indicate the orientation), and Fig. 6 shows the structural descriptions derived for these sets of lines in terms of the relationships described above. In theory, this structure should not change *too much* with the viewing

---

[3]The contrast is defined by the difference between the mean gray-level values associated with the left and right stripes.

point. In practice, for various reasons this is not true. It is therefore desirable to devise a matching strategy allowing for slight left/right dissimilarities.

Finally, Fig. 2 shows that the structural description advocated in this paper is not limited to the blocks' world.

III. GEOMETRIC CONSTRAINTS FOR MATCHING

In this section we consider the geometry of the stereo sensor. For a line in one image we determine the range of possible positions and orientations of its potential matches (assignments) in the other image. Concerning the position, a constraint is derived from the geometry of the sensor. Concerning the orientation, we show that the results obtained by Arnold and Binford [2] can be extrapolated to deal with our more complex sensor geometry.

In the sequel we consider a line in the left image and derive constraints for its matching lines in the right image. This left-to-right process might well be applied right-to-left.

The geometry of the sensor is shown in Fig. 3. There are two coordinate frames associated with the two cameras. The z-axis is the optical axis and the image plane is parallel to the xy-plane at a distance $f = 1$ from the origin. The right frame differs from the left frame by a rigid transform which is composed of a rotation of angle ϕ about the common y-axis and a translation of vector $\overline{OO'}$ (the sensor's baseline). This transform can be represented by a four by four matrix, A (homogeneous coordinates):

$$A = \begin{pmatrix} \cos \phi & 0 & \sin \phi & b_x \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & b_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

We consider a scene point P and let X, Y, Z be its left-frame coordinates. Similarly, X', Y', Z' are its coordinates in the right frame. The two sets of coordinates are related by the formula:

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = A \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2)$$

P projects onto the left image plane at p with coordinates $x = X/Z$, $y = Y/Z$, and $z = 1$. Similarly, it projects onto the right image plane at p' with coordinates $x' = X'/Z'$, $y' = Y'/Z'$, and $z' = 1$. By replacing X', Y', and Z' by their expressions given by (2) and noticing that $X = xZ$ and $Y = yZ$, we obtain the position of p' as a function of x, y, Z, and the sensor's parameters ϕ, b_x, b_z:

$$x' = \frac{Z(x \cos \phi + \sin \phi) + b_x}{Z(\cos \phi - x \sin \phi) + b_z} \quad (3)$$

and

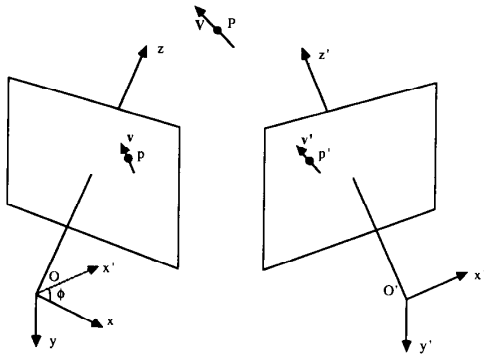$$y' = \frac{Zy}{Z(\cos \phi - x \sin \phi) + b_z}. \quad (4)$$

Fig. 3. The geometry of the two cameras.

Fig. 4. A top view of the epipolar plane defined by p, O, and O'.
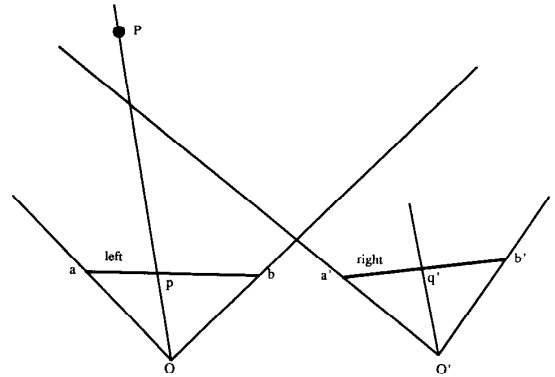
A. The Epipolar Constraint

We briefly recall the analytic formulation of the epipolar constraint. We eliminate Z in between (3) and (4). For a fixed left point p, and for a given sensor geometry we obtain a linear relation between x' and y':

$$yb_z x' + (b_x(\cos \phi - x \sin \phi)$$
$$- b_z(x \cos \phi + \sin \phi))y' - yb_x = 0. \quad (5)$$

This is the locus of the right image matches for a left image point and is called *the right epipolar line*. Similarly, one can determine a left epipolar line. Geometrically, this line is the intersection of the right image (or left image) with the plane defined by the points p, O, and O', i.e., *the epipolar plane*.

B. The Position Constraint

Now we analyze to what extent the location of p' may be constrained to belong to a certain segment along the right epipolar line associated with a fixed left point p. This is best shown in Fig. 4 which is a top-view of the epipolar plane passing by p, O, and O'. Necessarily, both P and p' belong to this plane. P is constrained to lie along the line Op. As P moves along this line, p' must lie somewhere in between a' and q', where a' lies onto the left border of the right image and $O'q'$ is parallel to Op. The position of q' can be derived analytically by letting Z tend

C. The Orientation Constraint

We consider a scene line passing through P with unit direction vector V. The coordinates of V are V_x, V_y, V_z and V_x', V_y', V_z', respectively. Let v and v' be the left and right projections of V. The left projection is given by the formula

$$v = k \times (V \times \overrightarrow{OP}) \quad (6)$$

where k is the unit vector associated with the z-axis. We obtain for the image coordinates of v:

$$v_x = ZV_x - XV_z = Z(V_x - xV_z) \quad (7)$$

$$v_y = ZV_y - YV_z = Z(V_y - yV_z) \quad (8)$$

$$v_z = 0. \quad (9)$$

The image slope of v is the ratio of v_y and v_x:

$$\tan \theta = \frac{V_y - yV_z}{V_x - xV_z} \quad (10)$$

where θ is the angle made by v with the x-axis. Similarly we can express the slope of v', i.e., $\tan \theta'$ as a function of x', y', V_x', V_y' and V_z'. Using matrix A, V_x', V_y', V_z' are expressed in terms of V_x, V_y, and V_z. Replacing x' and y' by their expressions given by (3) and (4) we finally obtain for the orientation of v':

$$\tan \theta' = \frac{\tan \theta (V_x - xV_z) \cos \phi + (yV_x - xV_y) \sin \phi + \dfrac{b_z}{Z} V_y}{V_x - xV_z + V_x\left(\dfrac{b_z}{Z} \cos \phi + \dfrac{b_x}{Z} \sin \phi\right) + V_z\left(\dfrac{b_z}{Z} \sin \phi - \dfrac{b_x}{Z} \cos \phi\right)}. \quad (11)$$

to infinity in (3) and (4). Hence, the segment $a'q'$ is the locus of possible positions of p'. When p is close to a, the segment $a'q'$ has its shortest length and when p is close to b, the segment $a'q'$ is as long as $a'b'$. This suggests that there are intrinsically less potential matches for left image points that are close to the left border than for points that are close to the right border! This is equally true for right image points close to the right border and it will have an important implication on the global strategy to be used for stereo.

This formula shows that the relation between θ' and θ is parameterized by the position of the line in the left image, x, y, the orientation of the line in the scene V_x, V_y, and V_z, the depth Z, and the parameters of the sensor b_x, b_z, and ϕ. The disparity in orientation $|\theta' - \theta|$ is the smallest for $x \to 0$, $y \to 0$, and $Z \to \infty$:

$$\tan \theta' = \tan \theta \cos \phi. \quad (12)$$

This equation determines a lower bound for the disparity in orientation. Let us try to determine an upper bound

for this disparity. The bound thus determined will correspond to the most defavorable case—the line is the closest to the sensor. Without loss of generality we consider the case $x = 0$, $y = 0$, that is, the position of the line in the left image is around the optical axis. Our sensor has the following characteristics: $\phi = 5.5°$, $b_x = 25$ cm, $b_z = 1.6$ cm.

We can now define a continuous function which maps points over the domain $V_x \times V_y \times V_z$ to pairs of image angles $\theta' \times \theta$. Given this mapping, we are able to translate probability distributions in one domain to the other domain. Assuming that the orientations of the scene lines have a uniform distribution (all orientations have equal likelihood) we want to establish whether some combinations of θ' and θ are more likely than others. Arnold and Binford [2] determined this probability distribution function. Their experiments led to the following results:

• For $\phi = 0$, $b_z = 0$, and $b_x/Z = 0.7$ they obtained a probability distribution function with a narrow saddle along the line $\theta' = \theta$ and half width at half maximum (hwhm) at the center was 30°.

• For $\phi = 0$, $b_z = 0$, and $b_x/Z = 0.07$ the saddle was even narrower and hwhm at the center was 3°.

• $\theta' = \theta$ is the most likely combination.

In our case $b_x/Z = 0.7$ corresponds to $Z = 35$ cm which is very close to the sensor. Given the characteristics of our sensor we can extrapolate Arnold and Binford's results. Hence, we have decided to bound the disparity in orientation by 0° and 30°.

IV. MATCHING AS GRAPH SEARCH

The main difficulty of stereo matching is that the two relational graphs derived from the two images are not identical. Indeed, the monocular structure described in Section II may change with the viewpoint for several reasons.

• A scene line visible in one image may be partially seen or may not be seen at all in the other image. This is due to occlusions, photometric variations between the two images, or to failures of the feature extraction process.

• A line in one image may appear broken in several pieces in the other image.

• Two lines separated in space may form a junction or be collinear in one image, and not in the other image. This is due to accidental alignment.

• The ordering constraint is sometimes violated. This may be due to transparent surfaces or to thin objects such as the legs of a chair.

The consequence of representing each image by a relational graph is that the stereo correspondence problem becomes the problem of finding a match between these two graphs. Since, for the many reasons listed above, these two graphs are not identical, the problem of matching them is a *double subgraph isomorphism* problem [5]. That is, the problem is to find all isomorphisms between subgraphs of a graph and subgraphs of the other graph. An isomorphism is a mapping function between the nodes

of the graphs which preserves internode compatibility. Such a mapping function has been defined in Section I-A.

Given two relational graphs we construct a correspondence graph (also called *association graph*) as follows. For each line in the left relational graph and in the right relational graph we construct a left-line-to-right-line assignment (node) in the correspondence graph. Then we connect two assignments whenever they are compatible. The best match could well be taken to be the largest set of assignments that were all mutually compatible under the relations. In the correspondence graph this is just a totally connected set of nodes—a *clique*. A clique in a graph is a completely connected subgraph, i.e., each node in the clique is directly connected to all the other nodes in the clique. A *maximal clique* is a clique that cannot be extended to include other nodes of the graph. The *largest maximal clique* is the maximal clique containing the largest number of nodes.

Thus the best matches are determined by the largest maximal cliques in the correspondence graph. The advantage of forming this graph is that it is a simple pure graph-theoretic structure which is amenable to pure graph-theoretic algorithms such as clique finding [5]. Nevertheless graph construction is domain-dependent and it constitutes the main body of this section.

The graph construction method that we suggest starts with node building. Each node is classified according to its expected utility. This classification allows the elimination of unlikely assignments without any further graph-search process. Arc building takes into account both compatibilities and incompatibilities between nodes. One major difficulty with such a structural approach is the fact that some relations are missing from the relational graph. This is because the feature grouping process previously described is not perfect and because it is too costly to establish the relation between any two image features. Hence, it is not guaranteed that totally connected sets are systematically produced. In order to overcome this difficulty we suggest and implement a compatibility propagation technique.

As already mentioned, the notion of best match is associated with the largest maximal clique in the graph. There may be several largest maximal cliques in a graph. In practice we have noticed that a "large" maximal clique which is not necessarily among the largest ones, may sometimes correspond to the best match. Therefore we associate a benefit function with each maximal-clique candidate. The maximal clique which maximizes this benefit is selected as the best stereo correspondence.

A. Building the Nodes

The nodes of the correspondence graph are built on the basis of the geometric constraints presented in Section III. Each node has a benefit associated with it. This benefit plays two roles. The first role is to classify the nodes according to their expected utility. The second is to compute the benefit of a clique by summing up the benefits of the clique's nodes.

We consider a line in the left image. The epipolar lines associated with its endpoints define a region in the right image. For lines which are almost parallel to their associated epipolar lines, this region is widened "artificially" in a direction perpendicular to the line. The potential candidates to be assigned to this line must lie in this region and must satisfy the position constraint described in Section III-B. Moreover, for an assignment to be accepted, the orientation of the line candidates must lie within the limits derived in Section III-C ($0°$ and $30°$).

The benefit B associated with the assignment thus obtained captures a measure of similarity between the two lines within the assignment. This benefit sums up the difference in contrast, the difference in length, and the difference in orientation—e.g., Section III-C, and the difference in the number of relations that exist in between each line and the nearby lines. B is given by the formula

$$B_{node} = 1/4 \left(\frac{\min (C_l, C_r)}{\max (C_l, C_r)} + \frac{\min (L_l, L_r)}{\max (L_l, L_r)} \right.$$
$$\left. + \frac{\min (\theta_l, \theta_r)}{\max (\theta_l, \theta_r)} + \frac{\min (N_l, N_r)}{\max (N_l, N_r)} \right). \quad (13)$$

The number of relations between each line and its nearby lines is an important measure because it reflects the local density of the structural description. As already mentioned, the arcs in the correspondence graph are produced *directly* by comparing interline relations, and *indirectly* (by propagation) whenever these relations are missing. A clique with many arcs produced directly is a better clique than one with many arcs produced indirectly.

Experimentally we noticed that for a given left line, its assignment classified first using this benefit is the correct one in 75 percent of the cases. The assignment classified second is the correct one in 15 percent of the cases. Hence, the best two candidates contain 90 percent of the good assignments.

B. Compatibility and Incompatibility

We consider two assignments, l_i/r_a and l_j/r_b. l_i and l_j are lines from the left image and r_a and r_b are lines from the right image. We want to establish a set of rules which allows us to assert whether two such assignments are compatible or not. These rules are built on the basis of similarities between the left and right monocular descriptions produced by the feature grouping process. Let $m_{ia} = l_i/r_a$ and $m_{jb} = l_j/r_b$. We have:

Rule 1 (collinearity): m_{ia} is_compatible_with m_{jb} if the following proposition is true:

$$((i = j) \text{ and } (a \neq b) \text{ and } (r_a \text{ collinear_with } r_b))$$

or

$$((i \neq j) \text{ and } (a = b) \text{ and } (l_i \text{ collinear_with } l_j))$$

or

$$((i \neq j) \text{ and } (a \neq b) \text{ and } (l_i \text{ collinear_with } l_j))$$
$$\text{and } (r_a \text{ collinear_with } r_b))$$

Rule 2 (connectivity): m_{ia} is_compatible_with m_{jb} if the following proposition is true:

$$(i \neq j) \text{ and } (a \neq b) \text{ and}$$
$$(l_i \text{ same_junction_as } l_j)$$
$$\text{and } (r_a \text{ same_junction_as } r_b) \text{ and}$$

(the angle made by l_i and l_j has the same sign value as the angle made by r_a and r_b).

Rule 3 (generalized ordering): m_{ia} is_compatible_with m_{jb} if the following proposition is true:

$$(i \neq j) \text{ and } (a \neq b) \text{ and}$$
$$(((l_i \text{ left_of } l_j) \text{ and } (r_a \text{ left_of } r_b)) \text{ or } ((l_i \text{ right_of } l_j)$$
$$\text{and } (r_a \text{ right_of } r_b))).$$

Rule 1 reflects the descriptive property that two collinear space lines are collinear in both images. It also embeds the fact that a line in one image may match a line in the other image that has been broken into two or more pieces. This is the only exception allowed with respect to the uniqueness-of-match constraint.

Rule 2 reflects the descriptive property that two or more lines intersecting in space intersect in both images.

Rule 3 is a generalization of the ordering constraint. Previous work has investigated this ordering constraint in conjunction with the epipolar constraint. Here, the ordering is introduced as an inherent consequence of the local photometric and topological similarities between the two images on the premise that we deal with opaque surfaces.

To summarize, these three rules express the compatibility between assignments on the basis of three descriptive properties: collinearity, connectivity, and ordering. One may now formulate *incompatibility* between matches:

Rule 4 (incompatibility): m_{ia} is_incompatible_with m_{jb} if the following proposition is true:

$$(l_i, l_j \text{ are linked by one relation, and } r_a, r_b \text{ are linked}$$
$$\text{by a different relation) or}$$
$$((i = j) \text{ and } (a \neq b) \text{ and } (r_a \text{ not_collinear_with } r_b)) \text{ or}$$
$$((i \neq j) \text{ and } (a = b) \text{ and } (r_a \text{ not_collinear_with } r_b))$$

This rule also guarantees uniqueness of assignments outside the context of Rule 1.

To those, one may add a fifth rule necessary for establishing compatibility for node pairs whenever there is no direct relation:

Rule 5 (propagation): m_{ia} is_compatible_with m_{jb} if there is a match $m_{kc} = l_k/r_c$ such that the following proposition is true:

$$(m_{ia} \text{ is_compatible_with } m_{kc})$$
$$\text{and } (m_{kc} \text{ is_compatible_with } m_{jb})$$

To conclude, the first four rules express the compatibility/incompatibility between matches established on the basis of the direct (measurable) relations between one feature and its nearby features. The fifth rule extends the local compatibilities to larger structures.

C. The Matching Strategy

We consider an example and show how the stereo correspondence problem is actually cast into a graph representation and how the best available solution is found. Fig. 5 shows a set of six left image lines (l_1-l_6) and a set of nine right image lines (r_a-r_i). Fig. 6 shows the left and right associated structural descriptions. The left and right structures to be matched are not identical. Part of the structure is occluded on the left side. Some lines are broken into pieces in one image and not in the other image. Clearly, the two structures are not isomorphic. We describe a left-to-right matching process:

1) Individual nodes are built. This is done by establishing a list of matches for each line in the left image. The resulting nodes are shown in Fig. 7, where m_{ij} corresponds to l_i being matched with r_j. A benefit B is computed for each node. This benefit takes the value 1 for a *perfect* match and tends to 0 as the dissimilarity increases. Moreover, for each left line, its associated assignments are classified on the basis of the value of B. Among these nodes, those with a benefit which falls below a locally computed threshold are eliminated. In any case, the best two nodes are always kept in the graph.

2) Rules 1, 2, and 3 are applied. The arcs linking compatible nodes are shown in Fig. 7. There are three isolated nodes and one connected graph component. This connected component cannot be interpreted directly in terms of a stereo correspondence between the two structures. Indeed, one can easily notice that within this connected component a left line still has several right line matches. Hence, the problem of stereo matching cannot be cast into the problem of finding connected components in a graph.[4] A correct solution may be found if one goes one step further and makes explicit the set of nodes that *are not* compatible before trying to produce completely connected subgraphs.

3) Incompatibility is detected by applying the fourth rule. First we consider an example where the relation between two lines in the left image is not the same as the relation between their matching lines in the right image. For example, l_1 matches r_f and l_4 matches r_e. Nevertheless, while l_1 is *left_of* l_4, r_f is *right_of* r_e. Hence, the nodes m_{1f} and m_{4e} are incompatible. This is best shown in Fig. 8. In this figure the dotted lines link *incompatible* nodes.

We consider a second example. There is a relation in one image but there is no relation in the other image. For example l_2 matches r_g and l_6 matches r_a; l_2 belongs to the *same_junction_as* l_6 but there is no detected relation between
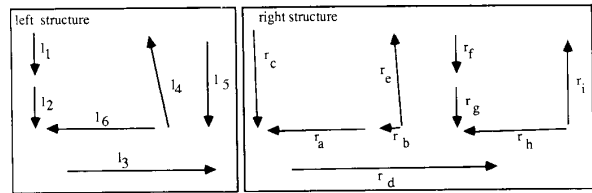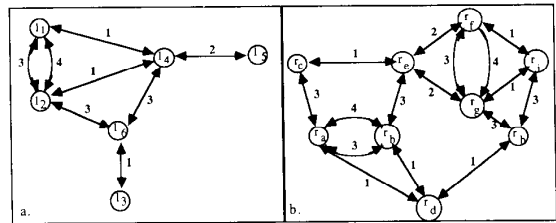

Fig. 5. Two images to be matched.

Fig. 6. Two structural descriptions to be matched. The interline relations are: left of (1), right of (2), same junction (3), and collinear (4).
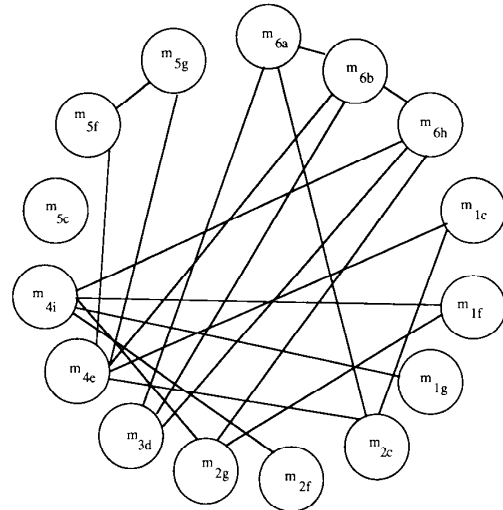

Fig. 7. The graph representation after node building and after applying the first three rules.

---

tween r_g and r_a. This relation was missed by the feature grouping process. The grouping process misses relations for two reasons: 1) grouping is concerned only with nearby features, and 2) it is influenced by noise and it may miss relations.

Therefore, we attempt to group these two features with more relaxed parameters than the parameters required during initial grouping. In this example, this second attempt failed and hence the relation between nodes m_{2g} and m_{6a} remains ambiguous. This ambiguity will be solved by the next step of the matching process.

4) Rule 5 is applied. The nodes are considered pairwise: if the pair is compatible or if it is incompatible, the graph is left unchanged. Otherwise, the propagation rule is applied.

5) Maximal cliques are searched. This is done using an algorithm described by Bolles and Cain [7]. Fig. 9 shows

---

[4]Searching for connected components is a linear function of the number of nodes and the number of edges of the graph being examined [21].
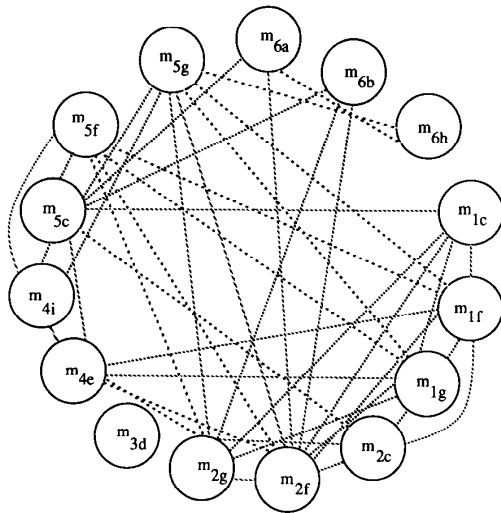
Fig. 8. The incompatible nodes found after applying the fourth rule.
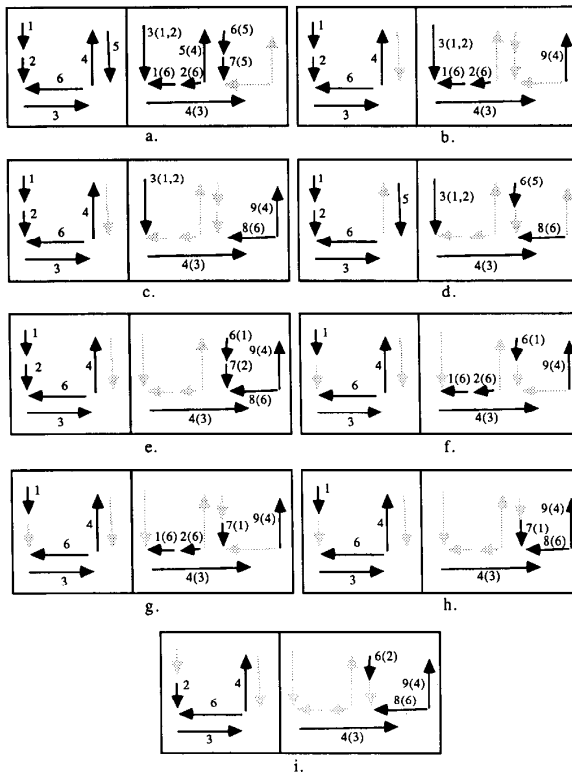

Fig. 9. All the matches corresponding to nine maximal cliques. The best solution (a) is the unique largest maximal clique found in the graph.

all the matches corresponding to nine maximal cliques found by this algorithm when applied to the example just described.

6) Each maximal clique is evaluated and the best one with respect to this evaluation (not necessarily the largest one) is equivalent to the best available stereo matching.

The evaluation of a maximal clique is simply the sum of its individual nodes' benefits

$$B_{\text{clique}} = \Sigma \, B_{\text{node}}. \qquad (14)$$

In the example described here, the best match corresponds to the unique largest maximal clique found in the correspondence graph.

D. Complexity

The complexity of the graph building process is proportional to the number of node pairs. One way to reduce this complexity is to carefully take into account the geometric and photometric constraints in order to eliminate as many incorrect left-line-to-right-line matches as possible.

The process of listing all maximal cliques in an undirected graph is known to be an NP-hard problem. Hence, all known algorithms to list maximal cliques are exponential in complexity [7], [5]. One way to increase efficiency is to maintain the number of nodes and arcs in the graph as low as possible. This may be done through detecting all possible incompatible nodes as described above. The detection of incompatible node pairs inhibits the creation of new arcs when the arc propagation rule is applied. Aside from the fact that efficiency is increased, this will reduce the chances of detecting false matches. Indeed, indirect compatibility created by propagation does not reflect measured image properties and hence, false matches tend to appear when this fifth rule is applied abusively. Hence, the detection of incompatibilities appears to be a key process.

V. Computational Experiments

The stereo matching strategy just described has been applied to a series of image pairs obtained with two cameras mounted on the INRIA mobile robot. Calibration data are available with these image pairs [12]. The images representing office scenes are shown in Figs. 10 (Office 1) and 17 (Office 2). The lines extracted from these images are shown in Figs. 11 and 18. The monocular descriptions resulting from the feature grouping processes are not shown. It is not realistic to apply the matching strategy directly to the entire image structures. Instead, we apply it locally to windows. A global match is obtained by applying the strategy to a partitioning of the images in terms of windows. Finally, the stereo matcher has been applied to an image pair representing a building roof, e.g., Fig. 21. The straight lines extracted from these images are shown in Fig. 22.

A. Local Matching

We apply the matching strategy to two 64 by 64 windows and to one 32 by 32 window. These windows are shown in Fig. 10 (left). Fig. 12 (left) shows 50 lines belonging to the first window. Fig. 12 (right) shows the assignments selected from the right image on the basis described in Section IV-A. There are approximately 3 right assignments for each line in the window. Next a graph is

Fig. 10. The first image pair. The three windows on which the local matching is demonstrated are shown on the left image.

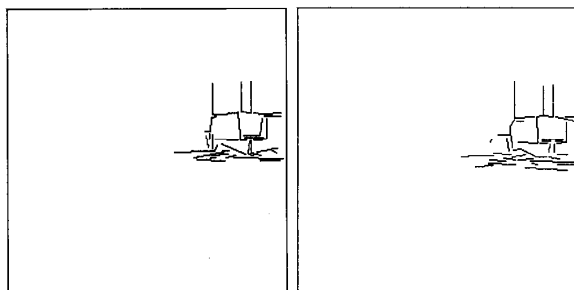Fig. 11. Lines extracted from the above images.

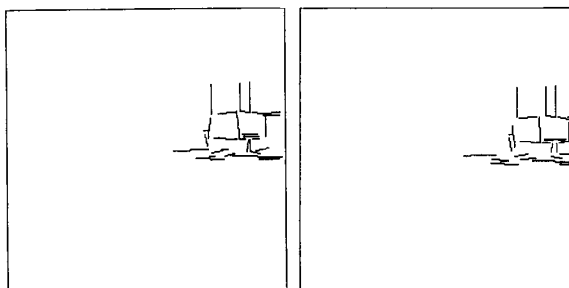Fig. 12. First example: candidates for matching.

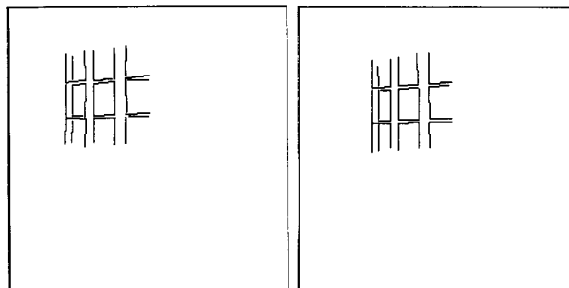Fig. 13. First example: the result of matching (35 lines).

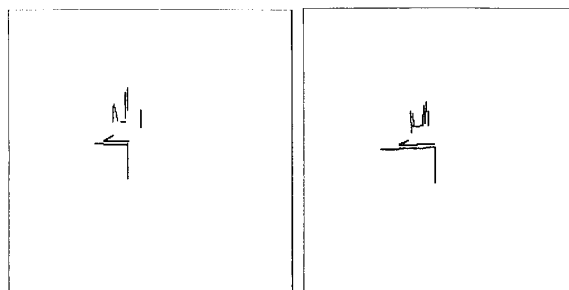Fig. 14. Second example: the result of matching (33 lines).

Fig. 15. Third example: the result of matching (10 lines).

Fig. 16. The matched lines for the first image pair.

built. After applying the five arc construction rules, we obtain a graph with 89 nodes and 1467 arcs (221 node pairs were found incompatible). Maximal-clique search is performed. Fig. 13 shows the best correspondence (35 left-line-to-right-line assignments) selected out of 4006 maximal cliques. This matching process (graph construction and search) took 174 seconds on a 11/780 VAX.

We applied the same strategy to the two next windows. In the second window there are 34 lines. The final graph has 74 nodes, 804 arcs and 307 incompatible node-pairs. The best correspondence has 33 left-line-to-right-line assignments selected out of 9 maximal cliques. See Fig. 14. This matching process took 14.5 seconds. One may notice that the computation time depends neither on the window size nor on the number of lines in the window. It depends on the local image structure and on the correctness with which this structure is detected. Table I summarizes the main results.

The final local matching example is shown in Fig. 15 and corresponds to a 32 by 32 window. In this case we obtain two disjoint graphs and hence two disjoint sets of

maximal cliques. The best correspondence comprises a four-node and a six-node maximal cliques.

B. Global Matching

We consider an image pair and a partitioning of the *left* image in terms of windows. Our current image size is 256 by 256 pixels and the current window size is 64 by 64 pixels. These windows are scanned one by one. Follow-

TABLE I
LOCAL MATCHING

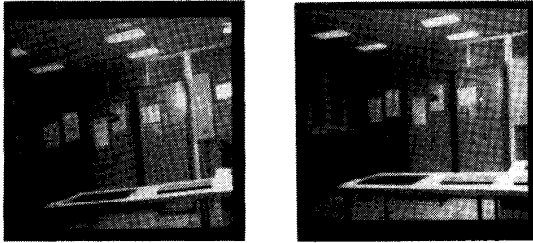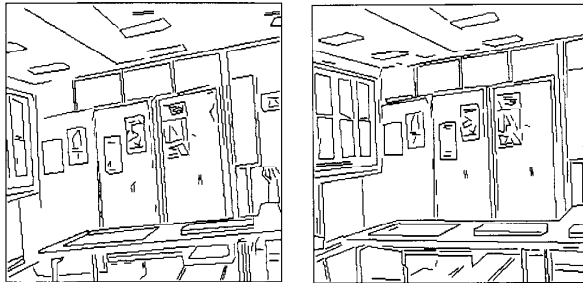| Local Matching	Window 1	Window 2
Window size	64 × 64	64 × 64
No. left lines	50	34
No. nodes	83	74
No. arcs	1467	804
No. maximal cliques	4006	9
No. final assignments	35	33
CPU time (seconds)	174.0	14.5

Fig. 17. The second image pair.

Fig. 18. Lines extracted from the second image pair.

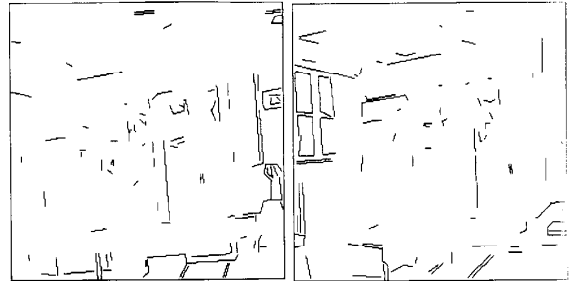Fig. 19. The matched lines for the second image pair.

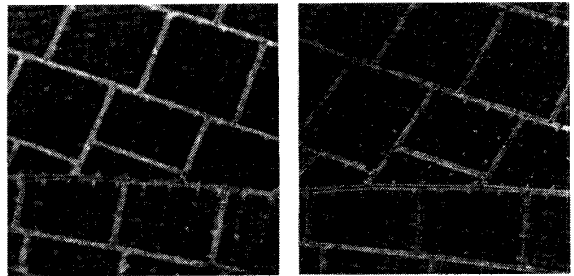Fig. 20. The unmatched lines for the second image pair.

Fig. 21. The third image pair.

ing the remark that the left side of the left image has less intrinsic matching candidates than its right side (Section III-B), the left image is scanned top-to-bottom and left-to-right.

The global matching strategy can be paraphrased as follows. All the lines crossing the current window are considered and the right line assignments are selected among the right lines that have not yet been included in a match. The graph associated with this set of assignments is built and all the maximal cliques are searched. The maximal clique best suited to represent the local stereo correspondence is selected.

Notice that with this approach a line may belong to two (or more) adjacent windows and hence it may have two different assignments: one assignment within the best maximal clique in one window and another assignment within the best maximal clique in the other window. These conflicts occur only for less than 10 percent of the lines which belong to several windows. We solve this type of conflict by selecting the assignment associated with the best correspondence.

This global matching strategy has been applied to the three image pairs shown in Figs. 10, 17, and 21. In the

first example (Office 1), starting with 425 left lines and 441 right lines, the algorithm found 295 left-line-to-right-line assignments and 9 conflicts. It successfully matched 70 percent of the lines in 26 minutes, e.g., Fig. 16. In the second example (Office 2), starting with 328 left lines and 316 right lines, the algorithm found 197 assignments and 6 conflicts in 6.5 minutes, c.f. Fig. 19. The unmatched lines are shown in Fig. 20. The unmatched lines are either too short to be significant or visible on one image and missing in the other image. In the third example (Roof) starting with 101 left lines and 122 right lines (Fig. 22) the matcher found 78 assignments and 8 conflicts. Fig. 23 shows the result of matching. In this figure matched lines have identical labels. For example, line 25 in the left image matched two collinear lines (both labeled 25) in the right image. These results are summarized in Table II.

VI. DISCUSSION

In this paper we suggested a method to solve for the stereo correspondence problem. The method consists of extracting local image structures and of matching similar such structures between two images. Previous approaches
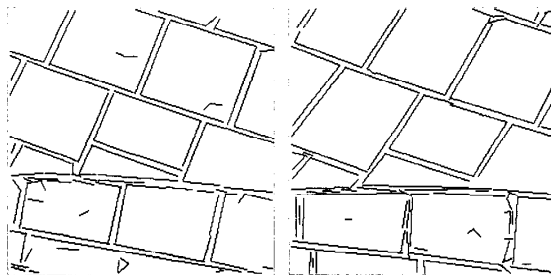
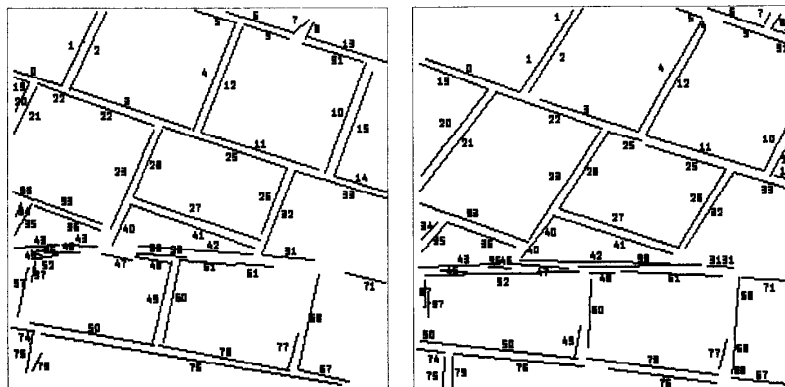Fig. 22. The lines extracted from the roof images.

Fig. 23. The matched lines for the roof images.

TABLE II
THE GLOBAL MATCHING

Global Matching	Office 1	Office 2	Roof
Image size	256 × 256	256 × 256	240 × 240
No. left lines	425	328	101
No. right lines	441	316	122
No. matches	295	197	78
No. bad matches	4	2	8
No. missed matches	26	36	15
No. conflicts	9	6	8
CPU time (minutes)	26.0	6.5	2.0
Improved time	4.4	2.0	0.3

have concentrated on matching individual features without exploring the constraints available with the relationships between them.

The matching itself is carried out by building a graph (an interpretation space) and by searching for the maximal cliques the best suited to represent a correspondence between the two images. This process is equivalent to looking for the best subgraph isomorphism between two relational graphs.

We propose an algorithm which uses straight lines and their relationships. Straight lines should be regarded as simple generic features. The method can be easily extended to other kind of features with their specific relationships. The matching itself can be utilized in the context of other problems such as the problem of matching an object model with image data.

The graph theoretic algorithm that we use guarantees that all the maximal cliques are found. This exhaustive

search explains both the quality of the results and the relative slowness of the method; it should not be compared to heuristic search methods. We prefer this algorithm because it allows an indepth analysis of the matching problem. One may solve for the maximal-clique finding problem using heuristics. As an example, we experimented a simple heuristic which gave results comparable to exhaustive search and which can be paraphrased as follows. The nodes of the correspondence graph are classified according to the number of their directly connected nodes. This strategy augments the chances of finding the largest maximal cliques before exploring the entire search space. Hence, one can stop the clique finding process once the algorithm has found a significantly large number of maximal cliques. The last line of Table II (improved time) lists the computation times obtained with this heuristic.[5]

Surprisingly enough, the method works well even if the images are "weakly" structured. The method fails to find the correct match when the real situation does not correspond to a situation predicted by the algorithm: violations of the orientation constraint, ordering constraint, accidental alignments, etc.

Our method is well suited for matching discontinuity edges and hence only a sparse depth map may be recovered. Therefore it should not be regarded as the unique way to solve for the stereo correspondence problem. We believe that it could be beneficial to combine this method

[5]All the computation times mentioned in this paper are those obtained with a 11/780 VAX.

with other methods which make use of different image cues such as texture, chromatic, and shading information.

We also plan to devise and implement a reconstruction procedure which will take into account the image structure and which will produce a 3-D description of the scene in terms of both a 3-D wireframe and piecewise continuous surfaces. The scene description thus obtained may constitute the input of higher level visual and action tasks.

ACKNOWLEDGMENT

REFERENCES

[1] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone, "A versatile computer-controlled assembly system," in *Proc. Third Int. Joint Conf. Artificial Intelligence*, Stanford Univ., Stanford, CA, Aug. 1973, pp. 298-307.
[2] R. D. Arnold and T. O. Binford, "Geometric constraints in stereo vision," in *SPIE Vol. 238—Image Processing for Missile Guidance*, 1980, pp. 281-292.
[3] N. Ayache and B. Faverjon, "Efficient registration of stereo images by matching graph descriptions of edge segments," *Int. J. Comput. Vision*, vol. 1, no. 2, pp. 107-131, 1987.
[4] H. H. Baker and T. O. Binford, "Depth from edge and intensity based stereo," in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, Vancouver, Canada, Aug. 1981, pp. 631-636.
[5] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[6] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Comput. Surveys*, vol. 14, pp. 553-572, Dec. 1982.
[7] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects, the local-feature-focus method," *Int. J. Robotics Res.*, vol. 1, no. 3, pp. 57-82, 1982.
[8] R. C. Bolles and R. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 3-26, Fall 1986; also in *Three-Dimensional Machine Vision*, T. Kanade, Ed. New York: Kluwer Academic, 1987, pp. 399-450.
[9] K. L. Boyer and A. C. Kak, "Structural stereopsis for 3-D vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 2, pp. 144-166, Mar. 1988.
[10] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vision*, vol. 1, no. 2, pp. 167-187, 1987.
[11] R. D. Eastman and A. M. Waxman, "Disparity functionals and stereo vision," in *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, Dec. 1985, pp. 245-254.
[12] O. D. Faugeras and G. Toscani, "The calibration problem for stereo," in *Proc. Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 15-20.
[13] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7 no. 1, pp. 17-34, Jan. 1985.

[14] M. Herman and T. Kanade, "Incremental reconstruction of 3-D scenes from multiple, complex images," *Artificial Intell.*, vol. 30, pp. 289-341, 1986.
[15] H. S. Lim and T. O. Binford, "Stereo correspondence: Features and constraints," in *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, Dec. 1985, pp. 373-380.
[16] G. Medioni and R. Nevatia, "Segment-based stereo matching," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 2-18, 1985.
[17] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 2, pp. 139-154, Mar. 1985.
[18] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Trans. Comput.*, vol. C-23, no. 8, pp. 860-870, Aug. 1974.
[19] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit," *Perception*, vol. 14, pp. 449-470, 1985.
[20] T. Skordas, "Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images," Ph.D. dissertation, Institut National Polytechnique de Grenoble, Oct. 1988.
[21] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Computation*, vol. 1, no. 2, pp. 146-160, June 1972.
[22] A. L. Yuille and T. Poggio, "A generalized ordering constraint for stereo correspondence," MIT Artificial Intell. Lab., Tech. Rep. AI Memo 777, May 1984.
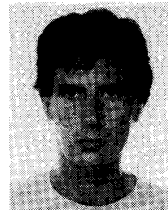
Radu Horaud was born in Bucharest, Rumania, on August 13, 1953. He emigrated to France in 1972. He received the "Diplôme d'ingénieur" from the Ecole Nationale Supérieure d'Ingénieurs Electriciens de Grenoble and the "Diplôme de docteur-ingénieur" from the Institut National Polytechnique de Grenoble, France, in 1977 and 1981, respectively.

He spent two years (1982-1984) as an International Fellow at SRI International in the Robotics Department. He is currently Chargé de Recherche at the Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle (LIFIA), Institut d'Informatique et de Mathématiques Appliquées de Grenoble (IMAG), France. His research interests include computer vision, computational geometry, 3-D object recognition and modeling, and parallel algorithms for image understanding.
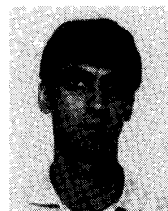
Thomas Skordas was born in Mylia, Greece, on February 13, 1962. He received the "Diploma in Electrical Engineering" from the University of Salonica, Greece, in 1984, and the "Diplôme de Doctorat en Informatique" from Institut National Polytechnique de Grenoble in 1988.

From 1985 to 1988 he was with the Laboratoire d'Electronique et de Technologie de l'Informatique (LETI), Grenoble, France. He is now with Industrie et Technologie de la Machine Intelligente (ITMI), Meylan, France. His research interests include computer vision, geometric reasoning, and graph theory.