Finding Geometric and Relational Structures in an Image

Radu Horaud, Françoise Veillon
LIFIA–IMAG, 46, avenue Félix Viallet
38031 Grenoble, FRANCE
and
Thomas Skordas
ITMI, Filiale de CAP–SESA, ZIRST Chemin des Prés
38240 Meylan, FRANCE

Abstract *We present a method for extracting geometric and relational structures from raw intensity data. On one hand, low-level image processing extracts* isolated *features. On the other hand, image interpretation uses sophisticated object descriptions in representation frameworks such as semantic networks. We suggest an intermediate-level description between low- and high-level vision. This description is produced by grouping image features into more and more abstract structures. First, we motivate our choice with respect to what should be represented and we stress the limitations inherent with the use of sensory data. Second, we describe our current implementation and illustrate it with various examples.*

1 Introduction and motivation

One of the fundamentals goals of computer vision is to interpret an image in terms of scene objects and relationships between these objects. One major difficulty associated with this task is that the image to be analysed and the objects to be recognized are embedded in two different representation frameworks.

On one hand, an image is a collection of pixels, each pixel being the 2D projection of a scene event. Hence, the task of the low-level processing is to make explicit the local image geometric and relational properties that are likely to represent local scene geometric structure.

On the other hand, an object is a collection of 3D primitives and relationships between these primitives. This representation must be transformed such that *visually salient* object features are made explicit. Then, the task of high-level vision is to match object descriptions against image descriptions.

In this paper we propose to investigate the image description the best suited for high-level tasks. We analyse the desired properties of this description from two viewpoints:

1. What *should* be represented in an image for successful scene interpretation?

2. What *could* be properly detected in an image?

Therefore our work is at the crossroads of low-level and high-level vision. We call this image description an "intermediate-level description," or InterLevel, because it embeds both feature detection and feature grouping. Feature detection is essentially a bottom-up process while feature grouping makes some assumptions about the geometry of the scene.

The description that we envision is a hierarchy of linear segments, relationhips between these segments, and more abstract geometric and relational structures. The latter are groups (collections) of segments satisfying some sets of properties. The InterLevel description currently produced by our system is best illustrated on Figures 1 and 2. Figure 1 (left) is the grey level image of a relatively simple object on a textured background. Figure 1 (right) shows the straight line segments detected in the previous image using a classical paradigm: edge detection, edge linking, and piecewise polygonal

approximation. The locations shown on Figure 2 (left) with a small circle are "terminations at a common point" or junctions. Segments and junctions are mapped in a graph. The image may be further searched for groups of segments satisfying various constraints. Figure 2 (right) shows some parallelograms and some curved contours (convex chains of connected, smoothly turning lines).

The remainder of this paper is organized as follows. Section 2 contains an overview of previous research efforts in the domain of feature grouping and structural description. Section 3 contains a description of the properties of the intermediate-level representation. Section 4 describes the current implementation. Finally we discuss the usefulness of our description with respect to ongoing research in computer vision.
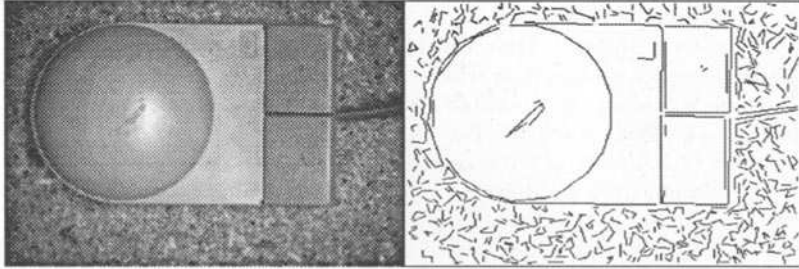

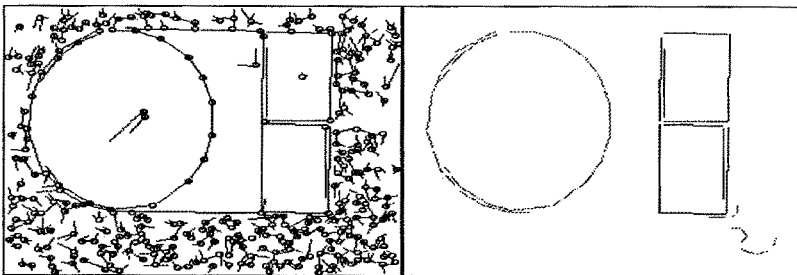Figure 1: A grey level-image and the detected straight lines

Figure 2: The relational graph extracted from the above image and some simple image structures

2 Background

The interest for feature grouping stems from Gestalt psychologists' figure/ground demonstrations, [13]: Certain image elements are organized to produce an emergent figure. Zucker, [13] distinguishes two types of grouping. The first corresponds to the inference of one-dimensional contours. The second involves orientation information that is densely distributed accross areas rather than curves. In this paper we are concerned with the first type of grouping.

Our work is also closely related to Marr's statement that an image contains two types of information [8]: Changes in intensity and image local geometry. Then, grouping is a process that makes both these pieces of information explicit.

Witkin and Tenenbaum [12] defined perceptual organization as a process able to detect *primitive structure*: Those relationships we are able to perceive even when we can't interpret them in terms of familiar objects. Our work may well be viewed as an attempt to build such primitive structures.

Connell and Brady [2] built a semantic network description of the image of a single object. The description contains both contour and region properties and relationships embedded in the "smoothed local symmetry" representation proposed by Brady and Asada [1]. The description thus

derived is used for learning visual object representations. With respect to the work of Brady and Asada, and Connell and Brady we advocate a description incorporating, among others, smoothed local symmetries. Unlike their semantic network, our graph contains lower-level information, i.e., primitive structure. Nevertheless we believe that the semantic network advocated by Connell and Brady can be inferred from our description.

Lowe [6] argued that perceptual groupings are useful to the extent that they are unlikely to have arisen by accident of viewpoint or position, and therefore are likely to reflect meaningful structure in the scene. Such groupings include collinearity, curvilinearity, terminations at a common point, parallel curves, etc. We will argue that the properties embedded in our description include (explicitly or implicitly) the groupings advocated by Lowe. Moreover, Lowe demonstrated the necessity of such groupings for object recognition.

From a more practical point of view, Dolan and Weiss [3] attacked the problem of perceptual grouping and implemented an algorithm which performed linking, grouping, and replacement in an iterative procedure. Sets of simple parametric tests were used to determine which structural relations were applicable to a given set of tokens. Then, a set of tokens was replaced by a simple token.

The description of an image in terms of a structural description has been proved to be useful for a variety of tasks. Shapiro and Haralick [10] introduced the idea of inexact matching between two structural descriptions. Inexact matching is an important concept. Consider for example the problem of stereo matching. The two images to be matched are different and hence, their associated structural descriptions are also different. Finding the best match between these two descriptions is an inexact matching problem. Skordas [11], and Horaud and Skordas [5] showed how to cast the stereo matching problem into a double subgraph isomorphism problem and how to solve it.

In conclusion previous research efforts have concentrated either on detecting groups of features in a complex image, or on proposing a structural image description, but not on both. In this paper we will argue that a representation associated with image structure is an inherent component of the grouping process itself. Unlike previous work we implemented a multilevel hierarchical structural description by which a group is built on top of less abstract groups.

3 Properties of the description

In this section we motivate the InterLevel description with respect to two main properties:

- its ability to constitute a reliable pointer into the memory of 3-D objects to be recognized, and

- its robustness with respect to the image formation process and the low-level segmentation process.

An image description in terms of its 2-D features is rarely an end in itself. Therefore, the vocabulary chosen for describing an image must contain those features and relationships that are likely to be associated with 3-D shapes. Moreover, among the properties of the latter, one must choose those which are invariant under projection.

Consider a 3-D object. In general, the boundary of such an object is a C^3 piecewise-smoothed surface. Unfortunately these surfaces do not appear directly in an image. Instead one has to deal with contours. There are two types of contours arising from surfaces: discontinuity and extremal contours. If two C^3 surfaces intersect such that at each point along the intersection the surfaces have distinct tangent planes, it can be shown that the intersection of the two surfaces is a C^3 edge. A discontinuity contour is the projection of such an edge. An extremal contour occurs whenever a smooth surface turns away from the viewer.

A point of intersection of two or more edges form an object vertex. The projection of edges and vertices give rise to image junctions which constitute important elements for describing an image. Interestingly, the range of possible configurations of contours and junctions is quite restricted. Malik [7] suggested an analysis based on the projection of edge and vertex neighbourhoods to predict the

possible image configurations. Figure 3 shows a catalog of image junctions, where each junction represents a combination of *linear* and *curved* contours. Then, the task of our system is to detect these junctions and to classify them accordingly.

discontinuity
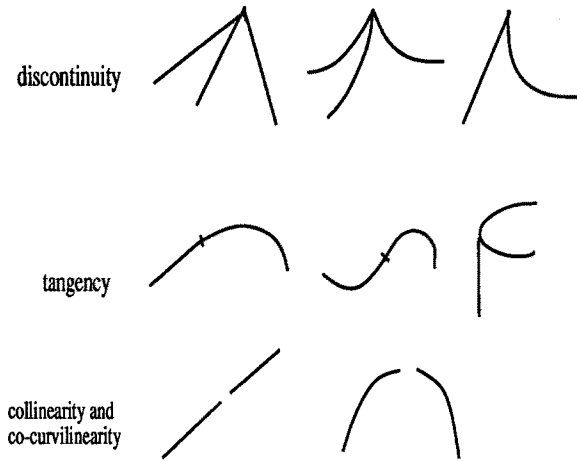
tangency

collinearity and
co-curvilinearity

Figure 3: A catalog of possible image junctions

Another important property encountered in many natural and man-made objects is local symmetry. The relationship between locally symmetric 3-D shape and image symmetry has been stressed by Marr [8], Horaud and Brady [4], and Ponce, Chelberg, and Mann [9]. It has been shown that under certain constraints object symmetry projects in the image to give rise to contour symmetry. Therefore, the detection and representation of image local symmetry is another important task of our system. Figure 4 shows a catalog of possible pairwise symmetric image contours.
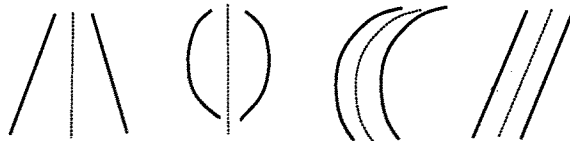
Figure 4: A catalog of possible image symmetric contours

Unfortunately, the accurate detection of contours and their relationships (junctions and symmetries) are affected by a certain number of undesirable and unavoidable phenomena.

First, the image formation process is a complex one and surface discontinuities are not the only scene features giving rise to contours. Color, textures, shadows, and specularities give rise to equally important[1] contours. We believe that the process of grouping, i.e., finding junctions and symmetries, partially throughs away uninteresting contours (contours that are not tight into a sought configuration). Hence, groups of contours are potentially more reliable than isolated ones. The "filtering" properties of grouping are stressed on the example shown on Figures 1 and 2.

Second, another unavoidable phenomenon is occlusion. For example, a circular edge at the intersection of two object surfaces may appear in an image decomposed in several pieces with relatively important gaps in between the pieces. This is the reason for which collinearity and co-curvilinearity are explicitly detected and added to our junction catalog, e.g., Figure 3.

---

[1]In terms of their image local structure.

Third, low-level image segmentation (feature detection) is known to be a non robust process. Figure 5 shows perfect junctions and symmetries and the configurations altered by segmentation.
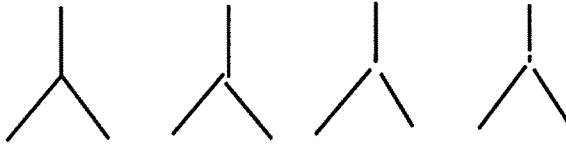

Figure 5: A "perfect" image junction and junctions altered by noise

To conclude this section, we have tried to build a collection of image features and relationships that are well suited for describing an image. In particular:

- Linear and curved contours, junctions and local symmetries are good candidates for describing 3-D shapes,

- Groups of image features are intrinsically less ambiguous than isolated ones. Therefore there will be a dramatic reduction of the complexity associated with image interpretation,

- The process of grouping eliminates undesired isolated features, and

- Noise and imperfect segmentation augments the difficulty of detecting groups of features. Hence, robust techniques are needed for feature grouping.

4 Implementation

We concentrate now on the process of detecting geometric and relational structures. This process takes as input the list of straight-line segments (shortly, segments) produced by the feature detection process and outputs a data structure which is the relational graph associated with the InterLevel representation. This graph is built on top of the initial list of segments and makes explicit relationships between these segments (junctions, collinearity, symmetry) as well as groups of segments satisfying some geometric constraints (curves, parallelograms, etc.). This graph is best illustrated in Figure 6. The graph is divided into two parts: The bottom level of this graph is constituted of a set of nodes where a node could be either a segment or an endpoint (as it will be explained later in this section, an endpoint is in fact a small region). Each segment has two endpoints associated with it. Two or several segments sharing a common endpoint form a junction. Junctions are represented implicitly in the bottom level of the graph or explicitly, in the top level. More generally, the top level graph contains such structures (i.e., combinations of nodes and arcs) as:

- junctions or termination at a common endpoint;

- curves which are sequences of connected segments;

- ribbons which are pairs of symmetric segments with the region in between them and

- other objects that can be built on top of these basic ones: pairs of symmetric curves, junctions combining curves and segments, parallelograms, sets of equally distant paraller lines or curves, chains of collinear lines, chains of co-curvilinear curves, groups of ribbons sharing a common symmetry axis, etc.

The organization of this graph is motivated by practical rather than theoretical considerations. Indeed, robust techniques are available for line detection. This is not the case for curve detection. A line has a simple analytical representation while a curve has not: A curve may be described by equations of various degrees. An alternative for describing a curve may well be "a smooth piece

P : point
S : segment
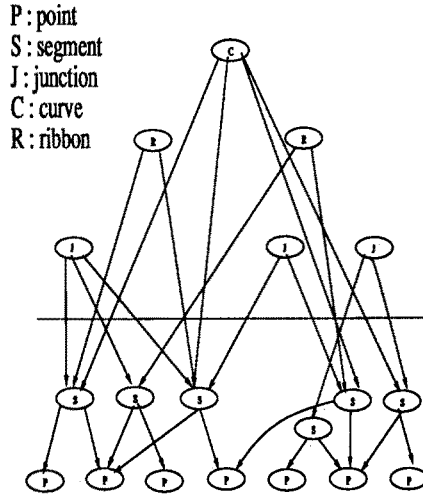J : junction
C : curve
R : ribbon

Figure 6: The InterLevel relational graph which describes image structure

of contour with no first or second order discontinuity." This is the description that we'll use in the graph. If specific knowledge about the exact shape of an object is available, one may take the Inter-Level curve description and fit it with the desired model. In the following paragraphs we describe in detail junction, symmetry, curve, parallelogram detection, and symmetry grouping.

4.1 Junction detection

An image junction is a set of lines terminating at a common point. In practice however, the lines rarely terminate exactly at the same point. Instead they terminate at a "small common region," i.e., Figure 7. Hence the difficulty of detecting junctions is twofold:

- the complexity of considering all subsets of lines as potential junctions, and

- the lack of a simple mathematical definition of a set of lines terminating at a common region.
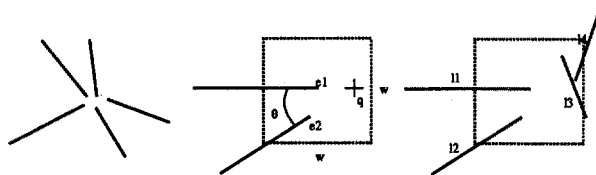
Figure 7: A junction is characterized by a set of lines terminating at a common region. The parameters of angle detection. A local configuration of segments.

In order to reduce the complexity mentioned above we build a simple data structure which consists of associating with each image pixel the information of whether there is a segment located at that pixel or not. If there is a segment, the data structure indexes it in the segment list. Then, for any image neighborhood one can quickly determine the list of segments with their endpoints terminating at this neighborhood.

It is easy now to determine the complexity of junction detection. Let n be the number of image lines and a be the average number of endpoints in the neighborhood of a line. The complexity is bounded by $2 \times a \times n$ since each segment has two endpoints. Next we describe a two-stage method

for detecting junctions: angle detection and angle grouping. Then we consider the special case of collinear lines.

Angle detection. An angle is a two-line junction. Consider an endpoint e_1 associated with a segment l_1 and a neighborhood of size $w \times w$ pixels centered at this endpoint. Let l_2 be another segment such that it has an endpoint e_2 lying in the neighborhood. Let q be the point of intersection of l_1 and l_2. Let d_1 be the distance from e_1 to q and d_2 be the distance from e_2 to q. Let θ be the value of the angle between the two segments. Segments l_1 and l_2 form an angle if the following conditions are satisfied, Figure 7:

1. $\epsilon_1 \leq \theta \leq \pi - \epsilon_1$

2. $d_1 \leq w/2$

3. $d_2 \leq w/2$

4. *q falls outside the segments l_1 and l_2*

5. *Test that there is no other line crossing $e_1 q$ and $e_2 q$*

Condition 1 guarantees that the lines are not collinear or do not form a very sharp angle (which is unlikely to represent a scene property). Conditions 2 and 3 guarantee that the point of intersection of the two considered segments lies within the "small region." For example, when applied to the configuration of Figure 7, only the pair l_1 and l_2 will be selected as an angle.

Angle grouping. Once all the angles have been detected, junction detection is quite straightforward. The principle of grouping is that two angles that share a common line and whose tips are not too faraway are fused into a unique junction. First a junction is initialised with an angle. Second the list of remaining angles is examined and angles are added, one by one, whenever they satisfy the above condition.

Detecting collinearity. Colinear lines are treated separately because one cannot always find the intersection of two collinear or almost collinear lines, i.e., Figure 8. Sometimes this intersection is simply not significant. Let again θ be the value of the angle between two segments, l_1 and l_2 and let e_1 and e_2 be their closest endpoints. Let d_1 be the distance from e_1 to l_2 and d_2 be the distance from e_2 to l_1. Let l be the amount of overlap of the two lines, i.e., Figure 8. The two segments are collinear if the following conditions are satisfied:

1. $\theta \leq \epsilon_1$ or $\pi - \epsilon_1 \leq \theta \leq \pi$

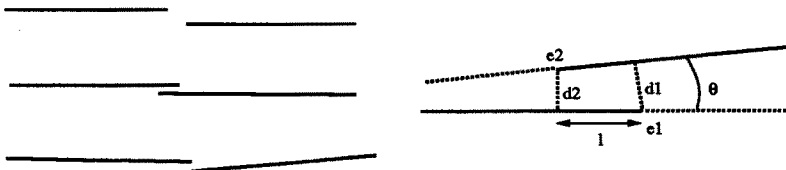2. $l \leq max\_overlap$

3. $\max(d_1, d_2) \leq w/2$

Figure 8: Pairs of "almost" collinear lines to be detected and the parameters of detecting these pairs.

4.2 Curve detection

Once the junctions are properly detected and mapped into the graph representation described above, one can build other image structures such as curves. A curve may be defined as a sequence of connected segments (the extremity of a segment and the origin of the next one in the sequence belong to the same junction). Let θ_i be the value of the angle between two consecutive segments in such a sequence. The sequence is a curve if the following conditions are satisfied:

1. $\epsilon_2 \leq | \theta_i | \leq \pi/2 - \epsilon_2$

2. $| \theta_i - \theta_{i+1} | \leq \epsilon_3$

3. $\theta_i \theta_{i+1} > 0$

4. *The curve is maximal*

Conditions 1 and 2 guarantee that the sequence turns "smoothly" and condition 3 gurarantees that there is no convexity change along the sequence. A maximal curve is a curve that cannot be extended to include another segment. The technique for detecting sequences with the above properties is based on a classical recursive graph traversal algorithm. This algorithm starts with a two-segment sequence and tries to extend it by exhaustively exploring the graph.

Let j be the number of junctions in the image and d be the average number of segments of each junction. The number of two-segment sequences is $j \times d \times (d-1)$. Hence the complexity of finding a curve with L segments is: $j \times d \times (d-1) \times (L-2) \times (d-1)$. In practice, d is bounded by 5 and the most likely value for d is 2. Therefore $d \times (d-1) \times (d-1)$ is bounded by a constant value.

4.3 Symmetry detection

Following the "smoothed local symmetry" definition in [1], the symmetry axis of two lines is the bisector of their angle. For two segments to be symmetric, they must have some amount of overlap. Referring to Figure 9, where l_1 is the length of one segment, l_2 is the length of the other one, and l is the length of their overlap, we must have:

1. $| l_2 - l_1 | \leq \epsilon_4$

2. $| (l_2 + l_1)/2 - l | \leq \epsilon_5$

Two segments together with the region in between them are called a "ribbon". One may further combine ribbons and build more complex symmetries.
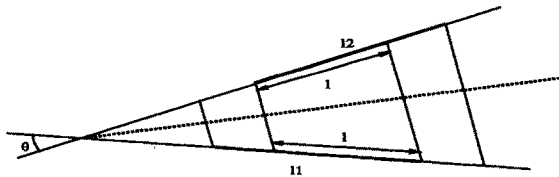

Figure 9: The definition of the local symmetry associated with two straight-line segments

4.4 Finding parallelograms

The task of finding all pairs of symmetric segments in the image is a costly one, i.e., proportional to the squared number of image segments. The detection of parallelograms is an example of a simple use of symmetry.
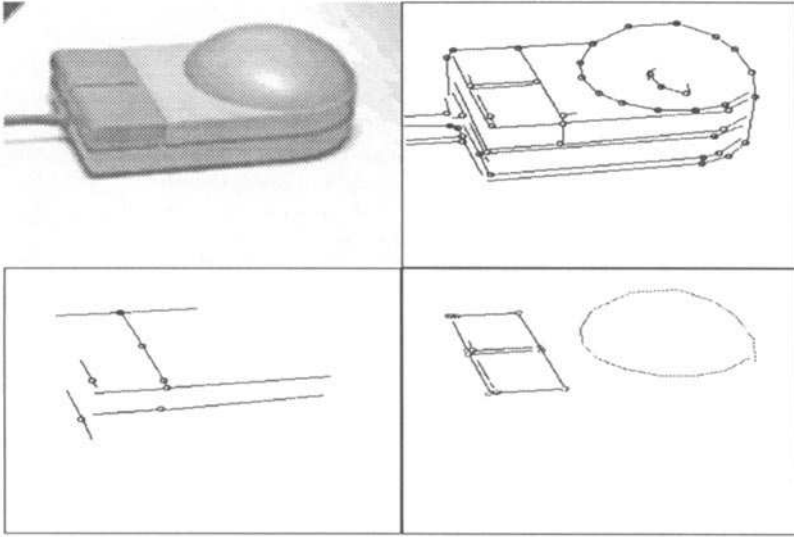
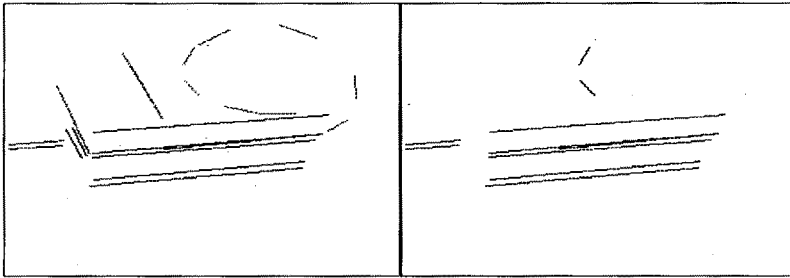Figure 10: Another image of the mouse and the obtained structures.

Figure 11: The pairs of symmetric lines (left) and a group of line-pairs with parallel symmetry axes (right).

We define a parallelogram as a sequence of four segments in the graph such that the first one is symmetric and parallel with the third one and the second one is symmetric and parallel with the fourth one. Notice that "symmetric and parallel" is just a particular case of the symmetry defined above. In our implementation we allow the side of such a parallelogram to be a meta-segment, that is, a sequence of collinear segments.

The technique for finding parallelograms is as for curve detection, a variation of a recursive graph traversal algorithm. For each junction and for each pair of segments (or meta-segments) within each junction we investigate all the segments (or meta-segments) that are connected to the initial pair and we select those four segments (or meta-segments) which could form a parallelogram.

To compute the complexity of this parallelogram search, let again j be the number of junctions and d be the average number of segments forming a junction. The number of segment pairs is (different than the number of two-segment sequences): $j \times d/2 \times (d-1)$. The cost of searching parallelograms is: $j \times d/2 \times (d-1) \times 2 \times (d-1) = j \times d \times d \times (d-1)$. Since d is bounded by a small constant value, the cost is a linear function of the number of image junctions.
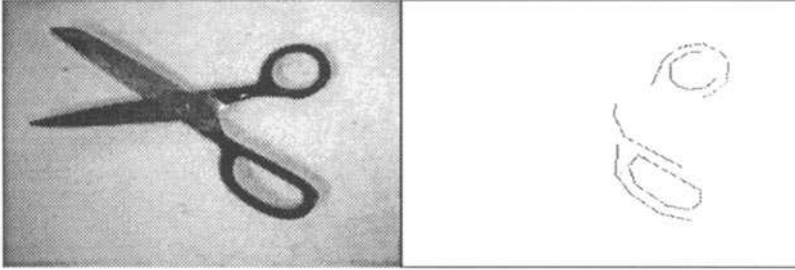
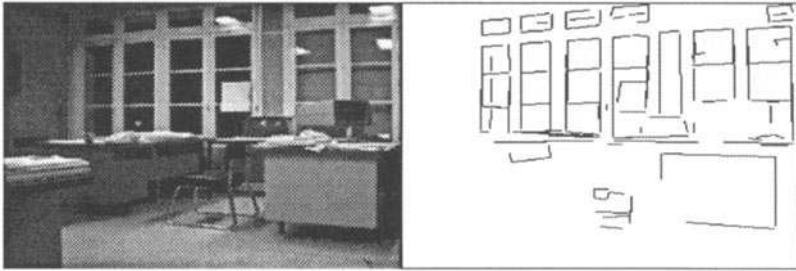Figure 12: The image of a pair of cisors and the detected curves.

Figure 13: The INRIA office image and the rectangles found by our system. Notice that the system have detected many rectangles that do not have a semantic interpretation.

4.5 Examples

We have run our system over a large database of images. The mouse of Figure 1 is shown from a different viewpoint on Figure 10 (top-left). There are also shown on this Figure the segment-and-junction graph (top-right), the sets of collinear segments, or meta-segments (bottom-left), and curves and parallelograms finally detected (bottom-right). Figure 11 shows all the pairs of symmetric lines (left) and a set of line-pairs with parallel symmetry axes (right).

Two more interesting examples are provided by the image of a pair of cisors, e.g., Figure 12 with the detected curves, and by the INRIA office image and the rectangles detected here, e.g., Figure 13. It is important to notice that the same processing (the same edge detector and line finder) with the same set of parameters have been used over the whole image set. It takes approximatively 10 seconds to entirely process a 256 by 256 image on a Sun 4/60: 8 seconds for edge detection and line finding and 2 seconds for grouping.

5 Conclusions

In this paper we presented a method for extracting geometric and relational structures under the form of a graph representation from raw intensity data. This graph is shown on Figure 6 and some image structures are shown on Figures 2, 10, 12, and 13. This graph may well be viewed as an intermediate representation between low- and high-level vision. The information embedded in the graph is useful for a variety of tasks.

Object recognition is often mapped into a graph matching problem: The sensory data is compared with object models. Currently we are investigating techniques for performing sub-graph isomorphism.

Another domain of computer vision for which our graph is a useful representation is visual learning. Consider for example the mouse object shown on both Figure 1 and Figure 10, and the graphs associated with these images. Is it possible, by comparing these two graphs, to derive a description

of the "mouse"?

Finally, we investigate techniques for combining the representation advocated here with camera motion in order to infer a 3-D version of the graph. This may be useful for modelling 3-D objects or for building scene descriptions in general.

References

[1] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, 3(3):36–61, 1984.

[2] J. H. Connell and M. Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[3] J. Dolan and R. Weiss. Perceptual Grouping of Curved Lines. In *Proc. Image Understanding Workshop*, pages 1135–1145, 1989.

[4] R. Horaud and M. Brady. On the Geometric Interpretation of Image Contours. *Artificial Intelligence*, 37(1–3):333–353, December 1988.

[5] R. Horaud and T. Skordas. Stereo Correspondence Through Feature Grouping and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1168–1180, November 1989.

[6] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publisher, 1985.

[7] J. Malik. Interpreting Line Drawings of Curved Objects. *International Journal of Computer Vision*, 1(1):73–103, 1987.

[8] D. Marr. Representing and Computing Visual Information. In Patrick Henry Winston and Richard Henry Brown, editors, *Artificial Intelligence: An MIT Perspective*, pages 17–82, MIT Press, 1979.

[9] J. Ponce, D. Chelberg, and W. B. Mann. Invariant Properties of Straight Homogeneous Generalized Cylinders and Their Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(9):951–966, September 1989.

[10] L. G. Shapiro and R. M. Haralick. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(5):504–519, September 1981.

[11] Th. Skordas. *Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images*. PhD thesis, Institut National Polytechnique de Grenoble, October 1988.

[12] A. P. Witkin and J. M. Tenenbaum. On perceptual organisation. In Alex B. Pentland, editor, *From Pixels to Predicates*, chapter 7, pages 149–169, Ablex Publishing Corporation, Norwood, New Jersey, 1986.

[13] S.W. Zucker. The Diversity of Feature Grouping. In Michael Arbib and Allen Hanson, editors, *Vision, Brain, and Cooperative Computation*, pages 231–262, MIT Press, 1988.