Segmentation and Grouping of Object Boundaries Using Energy Minimization*

Abstract

Traditionally, vision systems have largely relied upon the object boundaries extracted from images to accomplish recognition. Most contour detection algorithms, however, suffer from the fact that the extracted object boundaries are usually broken and incomplete due to poor imaging conditions and/or occlusions. In this paper, we describe an algorithm to perform curvilinear grouping of image edge elements for detecting object boundaries. The method works by generating hypotheses and selecting the best one. A neighborhood definition based on Delaunay graph is used to keep the number of hypotheses generated small. An energy minimizing curve is fit to the generated hypotheses to evaluate the grouping and locate discontinuities.

1 INTRODUCTION

Traditionally, vision systems have largely relied upon the object boundaries extracted from images to accomplish recognition. Recent psychophysical studies have provided evidence that contour-based processing also plays a very important role in object recognition by humans [4]. Thus, obtaining meaningful contours in degraded images is very important.

The main advantage of using a contour-based approach is the fact that most of the shape information about the object necessary for its recognition is contained in its boundaries. In addition, a great deal of three-dimensional (3D) inference can be made from just the two-dimensional (2D)

*This research was supported in part by NSF Grant no. CDA 8806599. This paper appeared in the *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 91)*, Hawaii, 1991.

contours. The human visual system seems to perform very reliably with stimulus-poor inputs. Line drawings are examples of such stimulus-poor inputs because they lack any of the usual surface related information such as texture, shading, range from stereopsis, etc. Even with degraded line drawings where the boundary information is incomplete, humans do a remarkable job by completing the missing boundary segments to recognize the objects.

Most contour detection algorithms, however, suffer from the fact that the extracted object boundaries are usually broken and incomplete due to poor imaging conditions and/or occlusions. In this paper, we describe an algorithm to perform curvilinear grouping of image edge elements for detecting object boundaries. One of the goals in this research is to have a unified computational framework that will complete piecewise smooth boundaries and detect discontinuities in curvature (corners) simultaneously. Another goal is to be able to apply the same methodology to integrate the results of various other modules such as symmetry detection with the results of curvilinear grouping module.

We start with a gray level image and we perform the low level processing of edge detection on it. Given noisy and broken object boundaries, we want to compute the best possible interpretation of these boundaries utilizing domain independent techniques. These boundary interpretations could then be input to higher level processes such as the 3D interpretation of the 2D lines [12, 19]. Current line labeling algorithms suffer from the fact that they expect perfect line drawings. We intend the contour completion and segmentation algorithm presented here to be one part of an integrated processing subsystem which will produce 3D interpretations of the objects in the scene possibly including a labeling of the lines. This point is very important because it means that our grouping algorithm does not have to generate perfect results by itself. It is sufficient for it to generate groupings that will start the processes in other modules going. In return, it can use partial results coming from other modules to perform further groupings. We do not address this problem in this paper, however, the results of the current algorithm must be evaluated in this context.

The paper is organized as follows. Section 2 gives a brief background on perceptual grouping and contour completion. Section 3 gives the details of the algorithm. Section 4 provides the experimental results and a discussion of these results. Section 5 gives some concluding remarks.

2 BACKGROUND

Perceptual grouping is the process that “organizes” image plane entities into larger units based on image plane relationships and Gestalt rules of organization. These Gestalt rules of organization are based on the premise that the physical world is well behaved and has structure. Such structure in the 3D world in turn is preserved during the imaging process so that the detection of significant structure (perceptual organization) in the image plane tells us something about the 3D structure of the scene [11]. It has been argued in the past that the human visual system uses the laws of perceptual organization to identify and complete the noisy contours of an object [3, 11, 22]. Such detection of structure by perceptual grouping is domain independent. That is, there are general rules reflecting the constraints of physical world and of the imaging process, but these constraints do not depend on the specific domain in which object recognition ultimately has to be performed. Gestalt psychologists were the first to state explicitly the rules of organization in visual perception. These rules include the properties of proximity, good continuation, closure and similarity [10, 21].

There has been a renewed interest in the perceptual grouping phenomenon both in the computer vision community [3, 6, 11, 20, 22] and the psychology community [14]. In these studies, the idea of detecting non-accidental properties has emerged as a unifying explanation for many of the phenomena that Gestalt psychologists had identified previously. Non-accidental properties in an image refer to properties such as collinearity, curvilinearity, parallelism, etc. The main idea is that the likelihood of such structures arising in images by accident is very low. Therefore, their presence in the 2D image indicates a significant causal structure that exists in the corresponding 3D scene. These non-accidental structures in images then must be identified at this intermediate stage which may be useful for processing at later stages.

The Gestalt rule of organization based on proximity and smoothness is one of the most important ones and has been used extensively in vision research. Even though these are usually stated as separate rules, in order to get any kind of reasonable groupings in images, they must be treated as a single rule in an integrated manner. This integration usually takes the form of a proximity-based grouping formulation with the smoothness criterion providing an extra constraint to resolve any ambiguities. Zucker and Hummel [23] and Ahuja and Tuceryan [3] have taken this approach.

Regularization is one method that has been suggested for enforcing smoothness criteria and thus restrict the possible solutions to various reconstruction problems [13]. This method has been successfully applied to various reconstruction problems [5, 9, 16]. Some of these applications perform very well, but require human intervention [9]. All of them have been used for various reconstruction tasks, but not necessarily for grouping applications. Some of them provide elegant algorithms for solving the variational problems that arise and for dealing with the discontinuities in a unified manner [5].

3 ALGORITHM

This section will present our formulation of the grouping of curvilinear contours by using an energy minimizing curve fitting technique. The top-level algorithm is given in Algorithm 1. The edge detector used for generating the initial edge map from the gray level image is the one by Canny [7]. We now describe each major part of the algorithm in more detail.

3.1 Preprocessing

In order to apply our grouping algorithm, the initial edge elements (edgels) should be sufficiently simple. Since the Canny edge detector already generates thinned edges, we do not need to perform any thinning operation at this stage. The remaining simplifications are that the edgels should not contain any branch points (any pixels which have three or more eight-connected neighbors) and that they should not contain any corners. The goal of the preprocessing step is to convert the initial edge data into a form suitable for further processing.

The preprocessing performs a connected components labeling on the edge map and identifies all the connected edge elements. For each component so labeled, it identifies branch points and breaks the edgel into smaller components at these branch points. The remaining edgels are further broken at points of high curvature (corners). The corner detection is accomplished by using the integrated algorithm of discontinuity detection and energy minimizing curve fitting described below (see Section 3.3). The second order derivative discontinuities detected in this scheme are labeled as corners. The edgels identified at the end of this process become the candidate image tokens to be grouped by our algorithm.

Algorithm 1

Input: Gray level image
Perform edge detection
$I_{edge} \leftarrow$ Result of preprocessing the edge image
Compute adjacency graph of the edgels in I_{edge}
E $\leftarrow$ Sort edgels in I_{edge} in descending order of length
G $\leftarrow \{\}$, the set of grouped edgels
foreach edgel $e \in$ **E** in descending order of length
    S $\leftarrow \{\}$
    $\mathbf{N}(e) \leftarrow$ set of neighbors of e
    foreach $e' \in \mathbf{N}(e)$
        $de \leftarrow$ Delaunay edge between e and e'.
        Construct a grouping hypothesis H consisting of the edgels (e, de, e')
        $U \leftarrow$ Fit an energy minimizing curve to H finding curvature discontinuities simultaneously
        S $\leftarrow$ **S** $\cup \{U\}$
    end
    Select from **S** the edgel that has no discontinuities and has the minimum energy.
    if the result is not empty
        remove the original constituent edgels and replace with grouped edgel.
    else
        remove edgel e from **E** and put into **G**
    end
end
Output: **G**, the set of grouped edgels.

end

3.2 Adjacency Graph

A crucial point in our algorithm is the definition of adjacency of image tokens. This is very important because if properly defined it reduces the number of the possible grouping hypotheses and also keeps the edgel relationships local. We use a modified version of the Delaunay graph as the basis of our adjacency definition.

It has been argued that the Delaunay graph can be used to represent local neighborhood information and that it is very useful in extracting local information about the spatial distribution of a set of points [1, 3]. It has also been shown that the Delaunay graph has desirable perceptual properties [8]. Alternative definitions of adjacency are possible. For a discussion of the advantages and disadvantages of various definitions see [1, 8]. We now give the definition of the Delaunay graph in terms of its dual, Voronoi tessellation. We then give the definition of our adjacency relation.

Let $\mathbf{S}$ denote a set of three or more points in the Euclidean plane. Assume that these points are not all collinear, and that no four points are cocircular. Consider an arbitrary pair of points P and Q belonging to $\mathbf{S}$. The bisector of the line joining P and Q is the locus of points equidistant from both P and Q and divides the plane into two halves. The half plane H_P^Q (H_Q^P) is the locus of points closer to P (Q) than to Q (P). For any given point P, a set of such half planes is obtained for various choices of Q. The intersection $\bigcap_{Q \in \mathbf{S}, Q \neq P} H_P^Q$ defines a polygonal region consisting of points in the Euclidean plane closer to P than to any other point. Such a region is called the *Voronoi polygon* [18] associated with the point P. The set of complete polygons for all points in $\mathbf{S}$ is called the *Voronoi diagram* of $\mathbf{S}$ [15]. The Voronoi diagram together with the incomplete polygons in the convex hull of $\mathbf{S}$ define a *Voronoi tessellation* of the entire plane. Two points $P, Q \in \mathbf{S}$ are said to be *Voronoi neighbors* if the Voronoi polygons of P and Q share a common edge. The *Delaunay graph* is obtained by connecting all the pairs of points which are Voronoi neighbors as defined above. The Delaunay graph is a planar graph and it triangulates the plane. It has also been shown that for a point pattern with a Poisson distribution the expected number of neighbors of a point in the Delaunay graph is a constant and is about six [2]. The fact that the expected number of neighbors of a point is constant keeps the number of hypotheses generated low.

The adjacency of the edgels is defined in terms of the original Delaunay graph computed for the edge map. We regard each edge pixel with row and column indexes (i, j) as a point in the plane and compute the resulting Delaunay graph. We then keep only those Delaunay edges which are connected to the endpoints of the edge elements. Thus, the adjacency of the edgels is defined only in terms of the adjacency of their endpoints. This is acceptable in view of the fact that we will be looking for curvilinear groupings along the boundaries.

3.3 Grouping by Energy Minimization

The heart of our grouping algorithm is the two step process of hypothesizing candidate groupings and picking the best groupings. The candidate groupings are generated by utilizing the adjacency graph. After each candidate hypothesis is generated, we fit an energy minimizing curve to the hypothesis. The optimal curve which is the result of this process will be smoothed, have curvature discontinuities marked, and have an energy value. The decision to group the edgels or not will depend on this resulting curve.

First, we describe the energy minimization formulation. Let the original pixels along the hypothesized edgel be given by the vector $\mathbf{x}(s) = (x(s), y(s))$ with parameter s. We would like to find a curve $\mathbf{u}(s) = (u(s), v(s))$ that minimizes an energy functional that depends upon the initial data $\mathbf{x}(s)$ and some regularization terms. The energy functional that we minimize is:

$$\begin{aligned} E \;&= \int_L \;\; \left\{ \lambda_1[(u(s) - x(s))^2 + (v(s) - y(s))^2] \right. \\ &\qquad \left. +\lambda_2[u_s^2 + v_s^2] + \lambda_3[u_{ss}^2 + v_{ss}^2](1 - k(s)) + \beta k(s) \right\} ds \end{aligned} \tag{1}$$

where $k(s)$ is an indicator function for the discontinuities along the curve given by

$$k(s) = \begin{cases} 1 & \text{if there is a discontinuity} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here u_s and u_{ss} are the first and second derivatives of $u(s)$ with respect to s, respectively. The notation is the same for $v(s)$. The λ_1, λ_2, and λ_3 are parameters that control the relative weights of the squared error and regularization terms in the energy functional. The $(u(s) - x(s))^2 + (v(s) - y(s))^2$ is the term that keeps the solution close to data. The term involving u_s and v_s is a regularization term ensuring smoothness in the first order derivatives. This can also be regarded as the stretch along the curve. The term involving u_{ss} and v_{ss} is a regularization term that enforces second order derivative continuity along the curve. If there is a discontinuity along the curve (i.e., a bend), then this term is turned off at that point. However, a penalty is paid for creating the discontinuity by an amount β. The second order regularization terms can be interpreted as the rigidity of the curve.

In order to obtain the discontinuities and the global minimum reliably, we have implemented this using the multiresolution algorithm described in [17]. We use three levels of resolution: course, intermediate, and fine. The grid size is halved each time going from finer to a courser resolution and vice versa. For each level of resolution, we compute the energy minimizing curve by the method described below and inject the results to the appropriate next level of resolution. Using the multiresolution method makes the computed shape of the curve and the locations of the second order discontinuities more reliable.

For a given grid resolution we discretize the energy functional in Equation (1) and we get:

$$\begin{aligned} E \; = \sum_{i=1}^{N} \; & \Big\{ \lambda_1[(u_i - x_i)^2 + (v_i - y_i)^2] + \\ & \lambda_2[(u_i - u_{i-1})^2 + (v_i - v_{i-1})^2] + \\ & \lambda_3[(u_{i+1} + u_{i-1} - 2u_i)^2 + (v_{i+1} + v_{i-1} - 2v_i)^2](1 - k_i) + \beta k_i \Big\} \end{aligned} \tag{3}$$

In the discrete case, the k_i are boolean variables that need to be computed. The curve fitting is done in two steps which is then iterated until the process converges on a solution. First, we estimate the discontinuities (i.e., the values of k_i) based on the current estimates of the curve (u_i, v_i). This is followed by the computation of the (u_i, v_i) values based on these discontinuity estimates by keeping them fixed.

The estimation of the discontinuities is done by minimizing the energy with respect to k_i. The expressions for k_i are given below [5]:

$$k_i = \begin{cases} 1 & \text{if } \beta \leq \lambda_3((u_{i-1} + u_{i+1} - 2u_i)^2 + (v_{i-1} + v_{i+1} - 2v_i)^2) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Once the dicontinuities are located according to Equation (4), keeping them fixed turns Equation (3) into a manageable minimization problem. The Euler equations (in discrete approximation) for solving this variational problem are as follows:

$$\begin{aligned} 0 &= \lambda_1(u_i - x_i) - \lambda_2(u_{i-1} + u_{i+1} - 2u_i) \\ &\quad +\lambda_3(1-k_i)(1-k_{i-1})(u_{i-2} + u_i - 2u_{i-1}) \\ &\quad +\lambda_3(1-k_i)(1-k_{i+1})(u_{i+2} + u_i - 2u_{i+1}) \\ &\quad -2\lambda_3(1-k_i)(u_{i-1} + u_{i+1} - 2u_i) \end{aligned} \tag{5}$$

$$\begin{aligned} 0 &= \lambda_1(v_i - y_i) - \lambda_2(v_{i-1} + v_{i+1} - 2v_i) \\ &\quad +\lambda_3(1-k_i)(1-k_{i-1})(v_{i-2} + v_i - 2v_{i-1}) \\ &\quad +\lambda_3(1-k_i)(1-k_{i+1})(v_{i+2} + v_i - 2v_{i+1}) \\ &\quad -2\lambda_3(1-k_i)(v_{i-1} + v_{i+1} - 2v_i) \end{aligned} \tag{6}$$

Equations (5) and (6) are then solved iteratively using the method described in [9].

The decision to group at the end will be made according to the following rule:

1. If there is a bend (a discontinuity) along the curve, the edgels are not grouped and this hypothesis is discarded.

2. If there is no discontinuity along the curve, then its energy is checked against the minimum seen so far. If it is less than the current minimum it is kept as a viable hypothesis. Otherwise, it is discarded.

At the end of this process, for a given edgel, we will either have a grouping hypothesis with the minimum energy, or we will have no hypothesis that survives. In the latter case we consider the edgel ungrouped and remove it from the list of edgels to be considered for further grouping. If there is a grouping found, then we remove the constituent edgels from the list of edgels to be grouped and add the new grouped edgel to the list. The entire process then continues from the start.

Since we start the grouping process with the longer edgels being considered first, then the most significant groupings will be formed first. Also, as long as the currently examined edgel can group with one of its neighbors, it will keep growing.

4 RESULTS

The algorithm is implemented in Common Lisp (the numerical routines are written in C which are called from Lisp) on a Sparcstation 1. In our experiments we have used the following values for the parameters: $\beta = .28$, $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 2.5$. These were determined empirically and seem to give reasonable results in most images.

We have tried our algorithm on a number of images. Two examples are given in Figures 1 and 2. The example in Figure 1 is an easy one and part (c) of this figure shows the resulting grouping. Note that the corners are detected reasonably well in this case and most of the groupings found are good.

Figure 2 was taken from [12]. As we can see most of the significant groupings have been formed by our program. Note that this result would be a big help for a line labeling program if it wants to generate some partial labelings for further processing. There are some places where the results can be improved. For example, the algorithm has difficulty detecting discontinuities where the corner forms an obtuse angle. The lower resolution grid finds this discontinuity, but when it is injected into the finer grid, it gets smoothed out and lost. This also causes problems with grouping because if the edgel is not broken at a corner, the proper hypotheses cannot be generated.

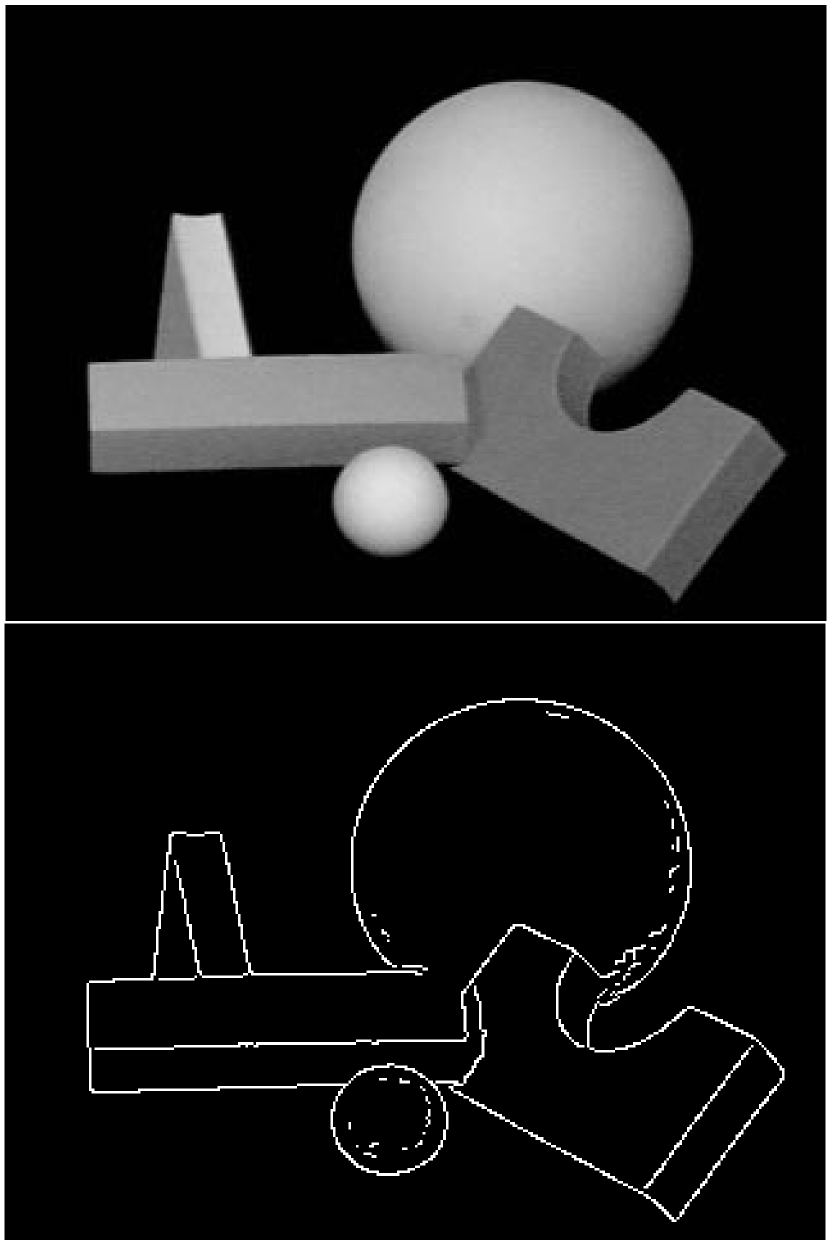
Also we have not integrated the lengths of the gaps into the energy minimization formulation. The stretch term may be used to deal with proximity based grouping in a unified manner.

5 CONCLUSION

Currently our grouping algorithm is able to detect most of the discontinuities in curvature, and it is able to form curvilinear groupings. Because we use the Delaunay graph as the basis of our adjacency definition, the number of grouping hypotheses generated are small.

Some aspects of the algorithm can be further improved. The algorithm currently is having difficulty detecting obtuse corners. This in turn causes problems with grouping because some of the candidate hypotheses cannot be generated. There is sufficient information in the low resolution grid of the energy minimization for doing this. We are currently working on this improvement.

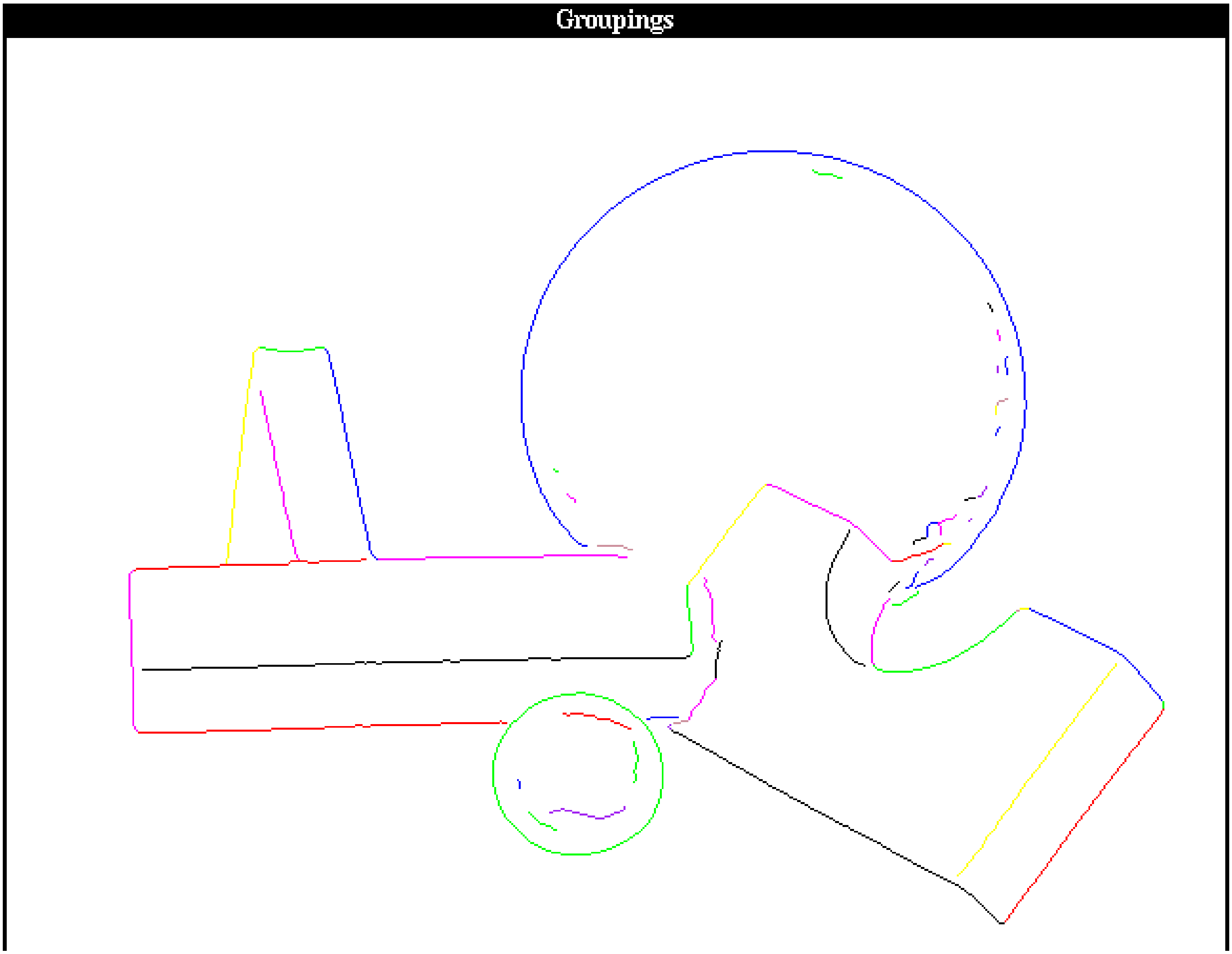
A second improvement of the algorithm is a sparse formulation where proximity based group-

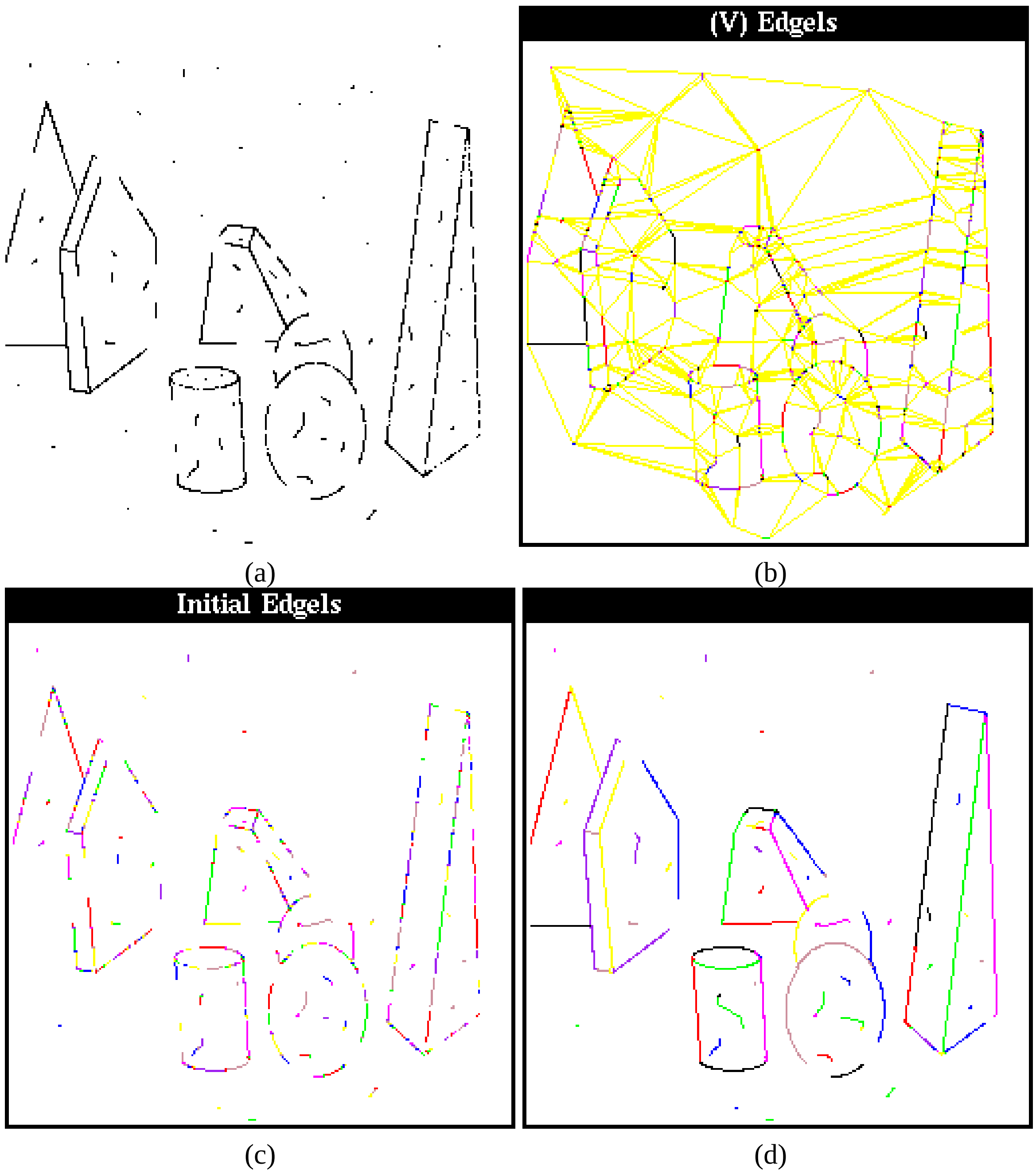


(a)

(b)

Figure 1: An example image and the groupings found. (a) Original gray level image. (b) The edge detector output. The resulting grouped edgels are shown on the next page. Adjacent edgels which are different are labeled with different colors.


Groupings



(a) (b) (c) (d)

Figure 2: An example image which has a large number of broken edge elements. (a) Original edge image. (b) The voronoi edgels (yellow) between edgel pixels. (c) The result of the preprocessing step which breaks the edgels at branch points and corners. Adjacent edgels which are different are labeled with different colors. (d) The resulting grouped edgels using the same coloring scheme.

ing can be computed automatically be creating discontinuities in the first derivative due to stretch. Our algorithm currently fills the gap between two edgels with the points along the Delaunay edge. An integrated treatment of this would be desirable. We are also looking into this aspect.

A further improvement is to integrate the curvilinear grouping module with other grouping modules such as symmetry detection. This type of integration would help to fill the side of the cylinder in Figure 2. A further step is to integrate the grouping module with 3D interpretation modules. In the context of 3D interpretation, some of the difficulties and ambiguities in the current algorithm may be resolved.

In conclusion, we intend this module to be part of a larger integrated system which includes other grouping modules as well as 3D interpretation modules. Therefore, the success of its results will depend on how helpful they are to the processing of other modules. Based on this, the grouping found in Figure 2 is an improvement because the line labeling module can now make some hypotheses about possible interpretations whereas the ungrouped image was almost intractable.

References

[1] N. Ahuja. Dot pattern processing using voronoi neighborhoods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(3):336–343, May 1982.

[2] N. Ahuja and B. Schachter. *Pattern Models*. John Wiley and Sons, Inc., 1983.

[3] N. Ahuja and M. Tuceryan. Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt. *Computer Vision, Graphics, and Image Processing*, 48:304 – 356, December 1989.

[4] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[5] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[6] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, 1986.

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, 1986.

[8] T. Chorzempa. Relative sensitivity of a family of closest point graphs in computer vision applications. Master's thesis, Michigan State University, East Lansing, Michigan, 1988.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321 – 331, 1988.

[10] K. Koffka. *Principles of Gestalt Psychology*. Harcourt Brace, New York, 1935.

[11] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academics, Boston, 1985.

[12] J. Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73–103, 1987.

[13] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, pages 314 – 319, 1985.

[14] I. Rock. *The Logic of Perception*. MIT Press, Cambridge, MA, 1983.

[15] M. I. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th Annual Symp. on Foundations of Computer Science*, pages 131–162, 1975.

[16] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24:52–96, 1983.

[17] D. Terzopoulos. Image analysis using multigrid relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):129–139, March 1986.

[18] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire: Recherches sur les parallélloèdres primitifs. *J. Reine Angew. Math.*, 134:198–287, 1908.

[19] D. Waltz. Understanding line drawings of scenes with shadows. In P. H. Winston, editor, *Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, NY, 1975.

[20] R. Weiss and M. Boldt. Geometric grouping applied to straight lines. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489–495, Miami Beach, Florida, 1986.

[21] M. Wertheimer. Laws of organization in perceptual forms. In W. D. Ellis, editor, *A Source Book of Gestalt Psychology*, pages 71–88. Harcourt Brace, New York, 1938.

[22] A. P. Witkin and J. M. Tenenbaum. On the role of structure in vision. In Jacob Beck, Barbara Hope, and Azriel Rosenfeld, editors, *Human and Machine Vision*, pages 481–543. Academic Press, New York, 1983.

[23] S. W. Zucker and R. A. Hummel. Toward a low-level description of dot clusters: Labeling edge, interior and noise points. *Computer Graphics and Image Processing*, 9:213–233, 1979.