Generic Edge Tokens: Representation, Segmentation and Grouping

Xiaofen Zheng
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, CA B3H 1W5
xzheng@cs.dal.ca

Qigang Gao
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, CA B3H 1W5
qggao@cs.dal.ca

Abstract

This paper presents a perceptual organization based method for describing, extracting and grouping generic edge features, called Generic Edge Tokens (GET). A GET is a perceptually significant image primitive which represents a class of qualitatively equivalent structure elements. A complete set of GETs includes both linear and non-linear segment classes and junctions of the segments. Edge traces extracted from image are segmented into GETs according to Gestalt Laws of proximity, continuity, and similarity in terms of descriptive image geometry and edge strength. In grouping 2D shapes, an object is described by its made-up GETs and the relations of GETs. The examples of ellipses and parallelograms groupings are provided as an illustration of the method.

1 Introduction

Object recognition is one of the most important fields in computer vision. Perceptual grouping, as its preprocessing, is the study of how features are clustered into perceptually significant groups for object recognition. The task seems very easy for the biological systems, yet it is very difficult to precisely formalize and describe these nature ability. Grouping is an extremely difficult problem from a computational point of view, and is certainly one of the least understood problem in vision [2]. The U.S. High Performance Computing and Communications initiative also identifies perceptual grouping as one of its four problem areas [9].

Despite the difficulties, in [12], various approaches of perceptual grouping to computer vision were summarized. Using perceptual grouping, Boldt [7] developed a bottom-up hierarchy of abstraction levels for collinear grouping of straight line segments. Based on proximity and good continuation in a local neighborhood, Dolan and Weiss [1] applied perceptual grouping to the extraction of curved lines. Rom and Medioni [10] provided a segmented axial description of a given shape by calculating the curvature changes. Mohan [8] suggested a method of locating collated features to describe 3D objects with particular shapes. In [13], 3D surface shapes were inferred from two kinds of 2D contours as parallel and skew symmetries. Saint and Medioni [11] presented a representation of edge contours extracted by symmetries and grouping operations. Kriegman [5] used a parameterized model for object recognition and positioning, and the parameters were derived from the theoretical contour and the observed data points. Most of these techniques were quantitative methods and involved intensive computation.

In this paper we present a generic approach for perceptual representation and grouping of 2D shapes. The robustness and efficiency of the method are demonstrated in the experiment results.

The paper is organized as the follows. Section 2 introduces perceptual organization and GET models. In section 3, the representation scheme and the grouping rules of ellipse and parallelogram are presented. Section 4 is the final conclusion.

2 GET Representation and Segmentation

2.1 GET Models

Generic Edge Tokens (GETs) are perceptually significant image primitives which represent classes of qualitatively equivalent structure elements. A complete set of GETs includes both linear and non-linear segment classes and junctions of the segments.

Gao and Wang [3] presented a curve partition method in a very generic form which allows the machine to perform a curve partition task following

the similar objectives in human visual perception. Generic segments (GSs) were used as the perceptual tokens. GSs can be tracked using tangents and traces of image edge curves. First, the initial edge pixels for tracking edges were determined based on the minimum dimension of the application objects, then the edge pixels along the traces were tracked according to the definitions of GSs. The detailed description of the algorithm can be found in [3].

A quantitative computation model of GS can be expressed as:

$$GS = \{p \,|\, \rho(p)\}$$

i.e., GS is a set of points satisfying some properties ρ. As shown in Figure 1 (b), generic segments are classified into eight categories according to the tangent functions of GS $y = f(x)$ and its inverse function $x = \varphi(y)$. The computative definitions for these generic segments are given in Table 1. We will give their detailed descriptions in section 2.2.

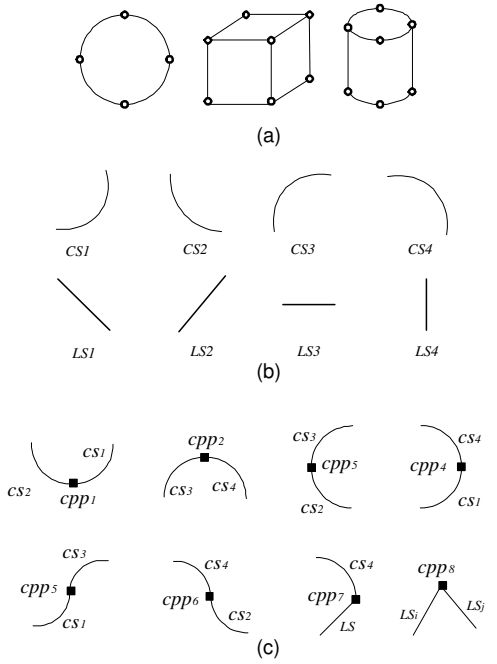

Figure 1: (a) The examples of curve partition. (b) The eight categories of generic segments. (c) The eight categories of curve partition points.

A curve partition point (CPP) is a perceptually significant point at which a transition of monotonicity takes place. The monotonicity is evaluated along a single segment in terms of monotonic increasing (denoted by "$M+$") or monotonic decreasing (denoted by "$M-$"). A CPP is the junction point at which two adjacent GSs are grouped into a structure

GS	$f(x)$	$\varphi(y)$	$f'(x)$	$\varphi'(y)$
$CS1$	$M+$	$M-$	$M+$	$M-$
$CS2$	$M-$	$M-$	$M+$	$M-$
$CS3$	$M+$	$M+$	$M-$	$M+$
$CS4$	$M-$	$M-$	$M-$	$M+$
$LS1$	$M-$	$M+$	c	c
$LS2$	$M+$	$M+$	c	c
$LS3$	c	n/a	0	∞
$LS4$	n/a	c	∞	0

Table 1: The definition of GSs. $M+$ and $M-$ stand for monotonic increasing and decreasing properties respectively.

which corresponds to a stable perception. As shown in Figure 1 (c), CPP are classified into eight general types as well. Their definitions are given in Table 2.

$Rule\#$	$Definitions$
$G1$	$(cpp_1, CS1, CS2)$
$G2$	$(cpp_2, CS2, CS3)$
$G3$	$(cpp_3, CS3, CS4)$
$G4$	$(cpp_4, CS4, CS1)$
$G5$	$(cpp_5, CS1, CS3)$
$G6$	$(cpp_6, CS2, CS4)$
$G7$	(cpp_7, CSi, LSj)
$G8$	(cpp_8, LSi, LSj)

Table 2: The definition of CPP.

2.2 GET Representation

GETs are the collections of the general feature classes GSs and CPPs. Figure 2 is the hierarchy structure of generic edge tokens. Here, $Tcpp$ means the true CPP and $Vcpp$ means the virtual CPP. $Tcpp$ is view invariant CPP. $Vcpp$ is sensitive to rotation change, but useful for grouping CSs to conic sections.

A general description of a segment is developed on the perceptual model of GS:

$$GSc\,[p_b, p_m, p_e]$$

where p_b represents the begin point of this segment, p_m for the middle point and p_e for the end point, c is the indicator signifying which category this segment belongs to. In this description (in Figure 3), the positions of the begin point p_b and the end point p_e of this segment are exchangeable.

For the curve segment (CS), the curvature information of this curve is characterized by its middle point p_m. While for straight line segment (CS),

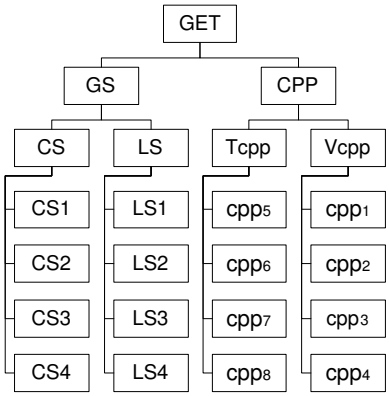

Figure 2: GET concept hierarchy.

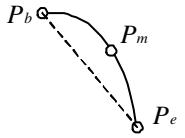

Figure 3: A GS.

three points p_b, p_e and p_m are in the same straight line, so the line can be determined only by its begin point p_b and end point p_e, and the middle point p_m is redundant. Therefore we set the value of p_m in straight line segments as $null$. The descriptions for the curve segment and straight line segment are respectively:

$$CSc\,[p_b, p_m, p_e]$$

$$LSc\,[p_b, null, p_e]$$

The general description of a CPP is represented as:

$$cpp_c\,[p, GSi, GSj],\ i \neq j$$

where p stands for the junction point, GSi and GSj represent different generic segments, and c denotes which type this CPP is. In this description (Figure 4), GSi and GSj share one end point p, i.e., $GSi\,[p, p_{m_i}, p_{e_i}]$, $GSj\,[p, p_{m_j}, p_{e_j}]$.

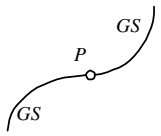

Figure 4: A CPP.

The general curve partitioning rules and grouping rules are both based on these generic perceptual tokens. All these tokens are perceptually distinguishable and can be defined qualitatively. They will be used as the basic elements of perceptual organization for qualitative analysis of different forms.

To describe an edge trace detected from image, we use a chain of GETs. The syntax representation of a trace is defined as:

$$T\,\{GS_{i_1}, cpp_{j_1}, GS_{i_2}, \cdots, cpp_{j_n}, \cdots\}$$

The compound 2D structures of (a), (b) and (c) in Figure 5 are described by their syntax representations respectively: $T_a\,\{LS3, cpp_7, CS2, cpp_5, CS3, cpp_2, CS4, cpp_4, CS1, cpp_7, LS3\}$, $T_b\,\{LS2, cpp_8, LS3, cpp_8, LS1, cpp_8, LS2, cpp_8\}$, $T_c\,\{LS4, cpp_7, CS2, cpp_1, CS1, cpp_7, LS4, cpp_7, CS1, cpp_1, CS2, cpp_7\}$.

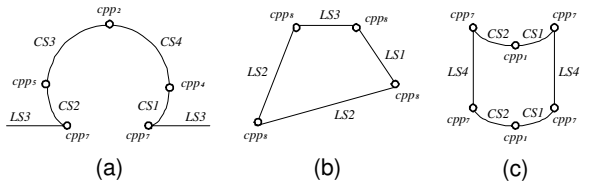

Figure 5: Trace examples.

A demonstration of GET segmentation is given in Figure 6, which shows the result CSs and LSs separately.

2.3 Segmentation

GSs are defined by the monotonic property of the curve segments in Table 1 and illustrated in Figure 1 (b). Figure 7 is a further graphic illustration of the definition of Table 1. For any curve segment, its monotonic change along X and Y axis are denoted as: $\triangle x_1, \triangle x_2, \cdots, \triangle x_n$ and $\triangle y_1, \triangle y_2, \cdots, \triangle y_n$ respectively, where P_0P_1, P_1P_2, $\cdots$, $P_{n-1}P_n$ are equally partitioned curve pieces which can be single point or multi-points interval lying on this segment. As we introduced, a general curve function and its inverse function are expressed as $y = f(x)$ and $x = \varphi(y)$. Their first derivatives are represented by $f'(x)$ and $\varphi'(y)$ respectively and their definitions are

$$f'(P_i) = \frac{\triangle y_i}{\triangle x_i}$$

$$\varphi'(P_i) = \frac{\triangle x_i}{\triangle y_i}$$

For instance, the curve segment in Figure 7 has the property: as $y_1 > y_2 > \cdots > y_n$, $f(x)$ is monotonic increasing; $x_1 > x_2 > \cdots > x_n$, so $\varphi(y)$ is

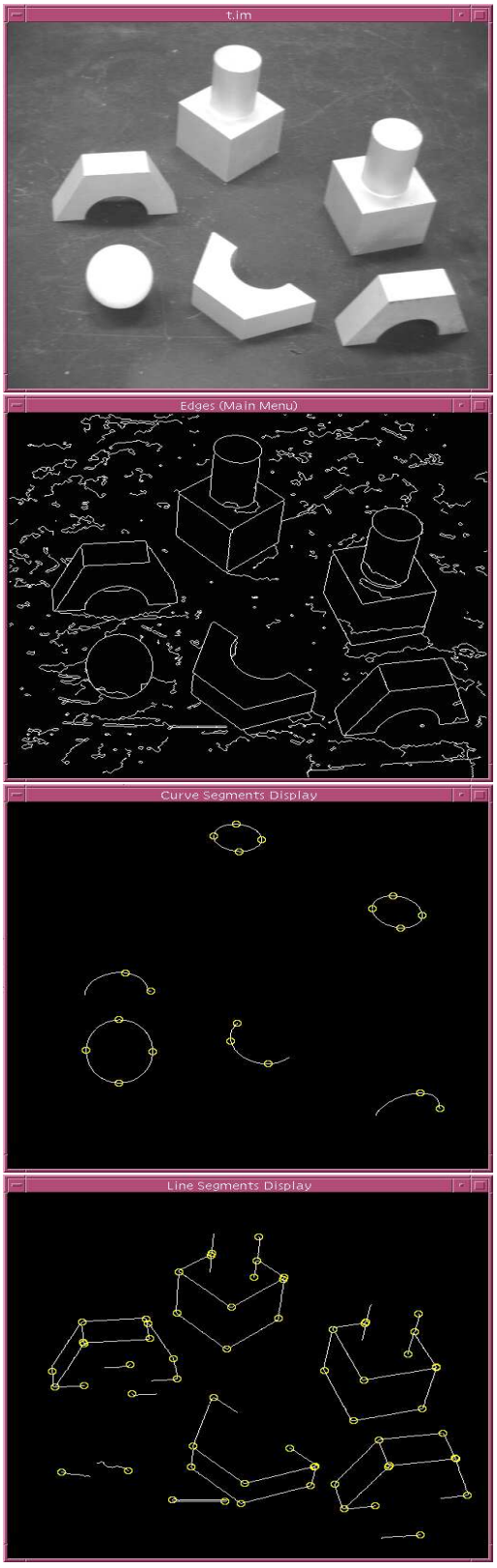

Figure 6: From the upmost to the bottom: (1) The original image; (2) Segments display; (3) Curve segment classes; (4) Line segment classes.

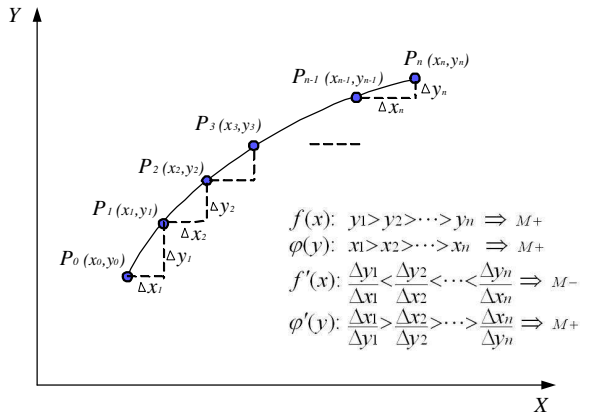

Figure 7: A graphic illustration of a curve segment.

monotonic increasing; $f'(x)$ is monotonic decreasing, i.e. $f'(P_1) < f'(P_2) < \cdots < f'(P_n)$, and $\varphi'(y)$ is monotonic increasing, i.e. $\varphi'(P_1) > \varphi'(P_2) > \cdots > \varphi'(P_n)$.

According all these properties, eight categories of generic segments are characterized so that GSs can be tracked qualitatively from image.

3 Perceptual Grouping

As Lowe [6] stated: There is a tendency for curves to be completed so that they form enclosed regions. To form a 2D shape, perceptual grouping takes place after GETs have been detected in an image. In the grouping process, cognition is not involved and it is unconscious.

Grouping processes rearrange the given data by eliminating the irrelevant data items and sorting the rest into groups, each corresponding to a particular object [4]. Grouping involves computationally intensive work of deriving structures from the images. GET based on perceptual organization can reduce the cost.

3.1 Ellipse Grouping

An ellipse is a closed planar curve which can be described qualitatively using GETs. Such a representation of ellipse is given in Table 3 which corresponds to the concepts illustrated in Figure 8 and Figure 9.

In Figure 9, an ellipse is represented by a graph in which each vertex stands for a CPP and each edge represents a GS.

As shown in Figure 8 and Figure 9, to group GETs into an intact ellipse, four types of curve segments and four types of CPPs must be identified. For incomplete matching, if there exists at least three

Ellipse {
$CS1\,[p_1, p_{m_1}, p_4]$
$cpp_1\,[p_1, CS1, CS2]$
$CS2\,[p_3, p_{m_2}, p_1]$
$cpp_3\,[p_3, CS2, CS3]$
$CS3\,[p_2, p_{m_3}, p_3]$
$cpp_2\,[p_2, CS3, CS4]$
$CS4\,[p_4, p_{m_4}, p_2]$
$cpp_4\,[p_4, CS4, CS1]$
}

Table 3: The definition of ellipse.

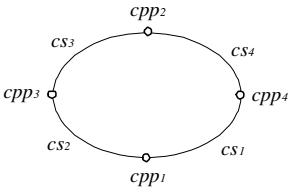

Figure 8: An ellipse model.

these CSs or three CPPs in a compound GET structure, these information gives viewers the significant meaning of an ellipse. So for any detected shape, we measure its confidence as:

$$Confidence = P_{Shape}\left(\frac{\sum GET_{Image}}{\sum GET_{Model}}\right)$$

where P_{Shape} means the certainty of the grouped shape could be, $\sum GET_{Image}$ is the sum of the GETs satisfied the structure shape from image, $\sum GET_{Model}$ is the total of GETs in the grouping model. The conditions of some possible incomplete matches are illustrated in Figure 10. For instance, a contour which has at least three structurally consistent CPPs or has at least three structurally consistent CSs is a strong indication that it is an ellipse. Therefore we can assume that we found an ellipse with the confidence measurement above 75%. This

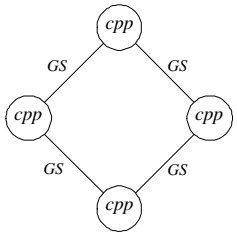

Figure 9: The graph representation.

tolerance is also useful in the recognition of partly occluded objects.

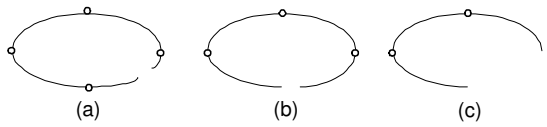

Figure 10: Some examples of possible incomplete ellipse models.

The ellipse grouping procedure is given in Algorithm 1.

Algorithm 1

1. Classify all CPPs into eight categories, and choose the categories cpp_1, cpp_2, cpp_3 and cpp_4 as the candidate groups;

2. Start from any given CPP $cpp_{b_1}\,[p_{b_1}, CS_i, CS_j]$ in candidate groups, according to the grouping rules, find the possible candidate CPPs in other candidate groups:

   (a) Along clockwise direction, find the possible adjacent CPPs sequence: $cpp_{b_2}\,[p_{b_2}, CS_j, CS_k]$, $cpp_{b_3}\,[p_{b_3}, CS_k, CS_l]$ and $cpp_{b_4}\,[p_{b_4}, CS_k, CS_i]$;

   (b) Along counterclockwise direction, find the possible adjacent CPPs sequence: $cpp_{b_4}\,[p_{b_4}, CS_k, CS_i]$, $cpp_{b_3}\,[p_{b_3}, CS_k, CS_l]$ and $cpp_{b_2}\,[p_{b_2}, CS_j, CS_k]$. This process will be stopped if any CPP was already detected in clockwise searching.

   (c) According to the confidence measurement to decide whether to save it as an ellipse.

CPPs provide strong evidence for grouping a meaningful objects. So using our generic model, algorithms based on recognizing from the CPPs are feasible as well and the task is to find the closed contours with four types of CPPs instead of finding four types of CSs. This flexibility is very helpful for the recognition of occluded contours in the future processing.

3.2 Parallelogram Grouping

Parallelogram is a four-cornered plane figure with opposite edges parallel. Mapping it to our perceptual tokens, the parallelogram (in Figure 11) consists of two pairs of parallel LSs and four CPPs which satisfies Table 4.

Parallelogram	{
	$LSi\,[p_4, N, p_1]$
	$cpp_8\,[p_1, LSi, LSj]$
	$LSj\,[p_1, N, p_2]$
	$cpp_8\,[p_2, LSj, LSi]$
	$LSi\,[p_2, N, p_3]$
	$cpp_8\,[p_3, LSi, LSj]$
	$LSj\,[p_3, N, p_4]$
	$cpp_8\,[p_4, LSj, LSi]$
	}

Table 4: The definition of parallelogram.

Figure 9 is also the graph representation of a parallelogram model. Both ellipses and parallelograms have the same graph representation. The difference lies in the types of the CPPs and GSs. This reduces the complexity of the graph representation scheme and provides the formalism on the construction of the graph model for the higher level of abstraction.

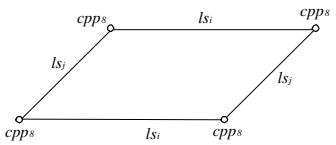

Figure 11: A parallelogram model.

To recognize a parallelogram, segments which are parallel with a certain tolerance ($\delta < 10^o$) are grouped as parallels. Segment pairs which meet these constraints generate parallels. For each LS, the possible candidate paralles are retrieved from the same group.

If given a contour which has at least three structurally consistent CPPs or has at least three structurally consistent LSs, there is a strong indication that that is a parallelogram and we assume we found a parallelogram with the confidence measurement above 75%. Therefore we can assume that we found a parallelogram. This tolerance is set for the recognition of partly occluded objects. Some possible incomplete matches are illustrated in Figure 12.

Based on the grouping rules, Algorithm 2 is an example for parallelogram recognition.

Algorithm 2

1. Classify all CPPs into eight categories, and choose the categories cpp_8 as the candidate group;

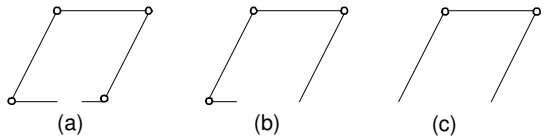

Figure 12: Some examples of incomplete parallelogram model.

2. Start from any given CPP $cpp_8\,[p_{b_1}, LS_i, LS_j]$ in candidate group, according to the grouping rules, find the possible candidate CPPs in this candidate group:

   (a) Along clockwise direction, find the possible adjacent CPPs sequence: $cpp_8\,[p_{b_2}, LS_j, LS_k]$, $cpp_8\,[p_{b_3}, LS_k, LS_l]$ and $cpp_8\,[p_{b_4}, LS_l, LS_i]$, and they must satisfy that LS_i is parallel to LS_k and LS_j is parallel to LS_l;

   (b) Along counterclockwise direction, find the possible adjacent CPPs sequence: $cpp_8\,[p_{b_4}, LS_l, LS_i]$, $cpp_8\,[p_{b_3}, LS_k, LS_l]$ and $cpp_8\,[p_{b_2}, LS_j, LS_k]$, and they must satisfy that LS_i is parallel to LS_k and LS_j is parallel to LS_l, This process will be stopped if any CPP was already detected in clockwise searching.

   (c) According to the confidence measurement to decide whether to save it as a parallelogram.

3.3 Experiments

The groupings of GETs for ellipses and parallelograms were demonstrated in Figure 13 and Figure 14 respectively. Each figure presents the original image, the GETs segmentation result from the image, and the final grouping result from the segmented GET map. The shape completeness measure was predefined with confidence $\geq$ 75% for both cases.

From the experiments, one may observe that in comparing with conventional equation based shape matching methods, the proposed method uses GETs as the minimum tokens for shape representation and grouping. Since GETs are descriptive in nature which can be manipulated qualitatively, therefore the main computation for grouping shapes can be reduced to binary reasoning. After GETs were segmented, they have been already classified into generic classes, so that the categorical heuristics can help to reduce the search effort for shape grouping

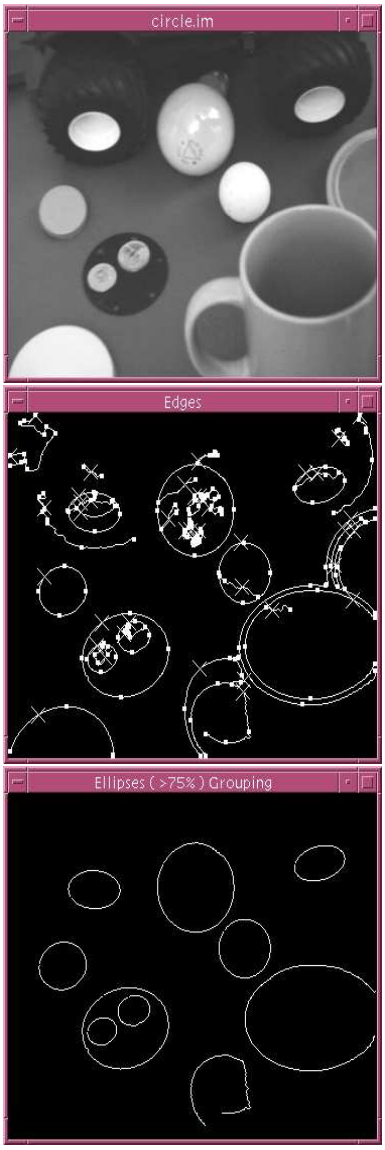

Figure 13: Top image: the original image; middle image: The edge detection and segments classification result; bottom image: the ellipses grouping result (confidence $\geq$ 75%).

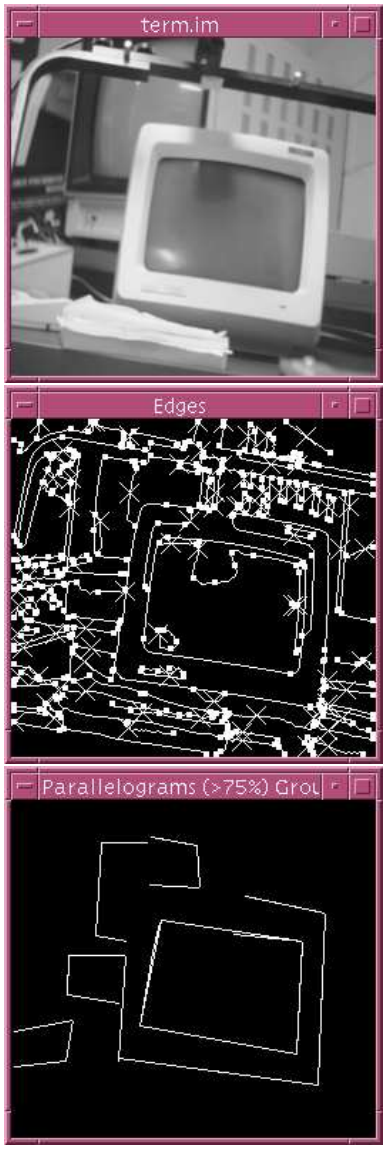

Figure 14: Top image: the original image; middle image: The edge detection and segments classification result; bottom image: the parallelograms grouping result (confidence $\geq$ 75%).

significantly. In the experiments, the grouping functions were implemented on SUN SPARC workstation, and the process were completed in 0.1 second.

4 Conclusions

GETs are perceptually significant edge features which are descriptive in the representation and therefore can be manipulated qualitatively in supporting perceptual grouping. GETs could also be a set of basic vocabularies of perceptual organization language for 2D shape representation and grouping. The low-level groupings of GETs are purely data driven, but do not depend on perfect edge data, so that they are very robust. The high-level shape groupings of ellipses and parallelograms are descriptive in nature. The principle computation for both levels of processes are qualitative in that no expensive mathematic models are involved. The results presented in the paper demonstrated that the computational advantages of applying perceptual organization in computer vision could be significant. The implementation of shape grouping presented in the paper was

based on the data in which GETs extracted from single edge traces. Future work will be extended to include grouping GETs extracted from different traces.

Acknowledgment

The authors gratefully acknowledge that this research received funding support from both NSERC and Deep Vision Inc. Deep Vision Inc. also provided the authors with their edge tracker software which was used in the research for producing original edge trace data.

References

[1] J. Dolan and R. Weiss. Perceptual grouping of curve lines. *Proc. DARPA Image Understanding Workshop*, 11(8):1135–1145, 1989.

[2] Jacob Feldman. The role of objects in percetual grouping. *Acta Psychologica*, 102:137–163, 1999.

[3] QiGang Gao and A.K.C. Wong. Curve detection based on perceptual organization. *Pattern Recognition*, 26(1):1039–1046, 1993.

[4] W.E.L. Grimson. *Object Recogniton By Computer: The Role of Geometric Constraints*. MIT press, first edition, 1990.

[5] D.J. Kriegman and J. Ponce. On recognizing and positioning curve 3-d objects from image contours. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(12):1127–1137, 1990.

[6] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):1121–1139, 1987.

[7] R. Weiss M. Boldt and E. Riseman. Token based extraction of straight lines. *IEEE Syst. Man Cybern*, 19(6):1581–1594, 1989.

[8] R. Mohan and R. Nevatia. Using perceptual organization to extract 3-d structures. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11(11):1121–1139, 1989.

[9] S. Negahdaripour and A.K. Jain. Challenges in computer vision: Future research directions. *IEEE Proc. Comput. Soc. Conf. Computer Vision and Pattern Recognition*, 92:189–198, 1992.

[10] H. Rom and G. Medioni. Hierarchical decomposition and axial shape description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):973–981, 1993.

[11] P. Saint-Marc and G. Medioni. B-spline contour representation and symmetry detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1191–1197, 1993.

[12] Sudeep Sarkar and Kim L. Boyer. Perceptual organization in computer vision: A reveiw and a proposal for classificatory structure. *IEEE Trans. on Systems, Man and Cybernetics*, 23(2):382–398, 1993.

[13] F. Ulupinar and R. Nevatia. Perception of 3-d surfaces from 2-d contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):3–18, 1993.