2D Geon Based Generic Object Recognition *

Xiangqian Yu, Vincent Oria
NJ Institute of Technology
University Heights
Newark, NJ 07102
{xy35,oria}@njit.edu

Pierre Gouton
Universite de Bourgogne
Aile Science de l'Ingénieur
21078 Dijon Cedex, France
pgouton@u-bourgogne.fr

Geneviève Jomier
Universite de Paris-Dauphine
LAMSADE, UMR CNRS 7024
place du Marechal de Lattre
de Tassigny75 775
Paris Cedex 16, France
genevieve.jomier@dauphine.fr

ABSTRACT

The Recognition by Components(RBC) is a theory in Psychology introduced by Biederman in the late 80s, by which humans perceive scenes through simple 3D objects with regular shapes such as spheres, cubes, cylinders, cones, or wedges, called Geons (geometric ions). Extracting geons from 2D images is a very challenging task as it requires a good segmentation and the recognition of the 3D geons in a 2D space. In this paper, we propose a novel approach for extracting 2D geons from 2D images. The process is composed of three major parts: image preprocessing which includes image background removal and segmentation, arc-geon detection, and polygon-geon detection. We also propose a general procedure for matching the extracted 2D geons to given models for object recognition. Experiment results show that our approach is competitive compared to existing object recognition methodologies in general.

Categories and Subject Descriptors

H.2 [**Media Analysis and Search**]: Miscellaneous—*Extract meaningful Geons from images and compare them with given models*

General Terms

Algorithm

Keywords

geon extraction, segmentation, levenshtein distance, object recognition

1. INTRODUCTION

The recognition of objects from images remains one of the most challenging problem in computer vision although it plays a significant role in intelligent systems(e.g. multimedia retrieval, robotics, etc). The previous works on object recognition can be grouped into two different categories: feature-based[5], and component-based as discussed in [1],[3]. Component-based have drawn much less attentions compared to feature-based approaches.

Shortly after the RBC theory[1] was proposed, several research projects were launched to find efficient and accurate approaches to geon extractions from images. The literature on geon extraction can also be divided into two categories: line drawing-based and image-based, depending on the type of input data.

The line drawing-based approach assumes that the contour information of the objects are given and focuses on identifying the geons. For example, Munk Fairwood [3] proposed an approach which uses logical program to extract geons from manually drawn objects.

The image-based approach aims at providing a full procedure of recognition with the entire image as input. For example, Du and Munck Fariwood[2] presented a feature grouping approach, which takes a set of straight lines collected from a segmented image to form geons. In [9], Wu and Levine used parametric geons as coarse descriptors of objects and defined seven 3D geon types.

Nevertheless, all those approaches have focused on very simple objects created from hand drawing or simple models, with monotonous backgrounds which makes them not applicable to general objects recognition in complex scenes. This paper defines a finite 2D geon set based on the parametric representation of 3D geons presented in [9], and proposes a systematic approach for 2D geon extraction. For object recognition purpose, we are proposing an object matching method in addition. The rest of the paper is organized as follows: Section 2 introduces 2D geons and the geon extraction procedure; Section 3 discusses how the geons are compared to given object models; Section 4 presents the results and Section 5 concludes the paper.

2. 2D GEON DEFINITION AND EXTRACTION

We first discuss the mapping of 3D geons into a 2D space, and then present the geon extraction procedure we are proposing.

2.1 Mapping 3D Geons into 2D Spaces

Parametric 3D geons are a finite set of distinct volumetric shapes, which are used to describe the shapes of object parts. Wu and Levines defined seven types of parametric geons according to equation.1. Fig.1(a) shows the parametric 3D geons defined in [9]. The parameter ϵ_1 defines the *squareness* in the north-south direction and ϵ_2 controls the *squareness* in the east-west direction. The parameters a_1, a_2, a_3 control

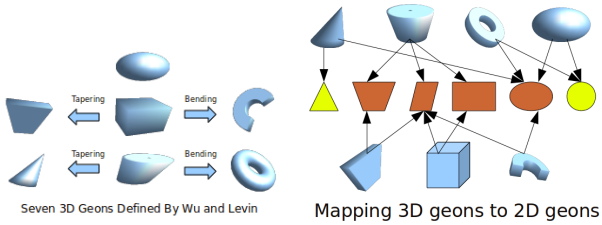
*Area Chair: Lexing Xie.

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

the scale of the 3D shape along x, y, z axes.

$$\left(\left| \frac{x}{a_1} \right|^{2/\epsilon_2} + \left| \frac{y}{a_2} \right|^{2/\epsilon_2} \right)^{\epsilon_1/\epsilon_2} + \left| \frac{z}{a_3} \right|^{2/\epsilon_1} = 1 \qquad (1)$$

A 3D geon needs too much information to identify while what an image contains is quite limited. We define six types of 2D geons from 3D parameter geons discussed above. Compared to a 3D geon, a 2D geon needs less information to identify. By projecting the 3D geons onto a 2D space, we get three basic 2D geon types: Ellipse, parallelogram, and trapezoid as shown in Fig.1(b). The other three extended types (circle, rectangle, and triangle) are special cases of the three basic ones. An ellipsoid can be mapped into an ellipse(circle is a special case of ellipse); a cylinder can be mapped into a parallelogram from the side-view (along Y-axes) or an ellipse from sky-view(along Z-axis); a tapered cylinder (a corn is a special case of tapered cylinder) can be mapped into a trapezoid (a triangle is a special case of trapezoid) from the side-view or an ellipse from the sky-view; a distorted cylinder can be mapped into part of an ellipse; a cuboid can be mapped into multiple parallelograms; a tapered cuboid can be mapped to a combination of trapezoid and parallelogram; and a distorted cuboid can be mapped to a combination of a partial-ellipse and a parallelogram.



(a) 3D Geons (b) 2D Geons

Figure 1: 3D Geons and Corresponding 2D Geons

2.2 2D Geon Extraction

We group the six types of 2D geon types into two categories: arc-geon, which includes ellipse and circle geons, and polygon-geon, which includes triangle, trapezoid, rectangle, and parallelogram. In order to extract 2D geons from an image, we first pre-process the image to remove the background pixels and segment the image into regions; then the arc-geons are extracted first during the arc-geon detection phase and detect all other geons in polygon-geon detection phase; and finally we validate all extracted geons.

2.2.1 Image Preprocessing

The image pre-processing comprises two parts: background removal and image segmentation. Since we are more concerned about objects which in most cases are in the foreground of an image, we first remove as many background pixels as possible. We use the background removal algorithm proposed by Lu and Gou[7]. The background in an image is defined[7] based on two hypotheses: 1) Background regions are normally connected to borders and are of relatively smaller sizes; 2) the regions occupying the corners are most likely to be part of the background.

We believe that a geon raw contour can be extracted through grouping the pixels in an image by their color or grey intensity which can be achieved through image segmentation. It is true that a perfect image segmentation algorithm remains to be found. Nevertheless, choosing a good segmentation algorithm is critical to the success of the geon detection. In this paper, we used Felzenszwalb's efficient graph-basic image segmentation[4]. It segments an image based on color-distances between pixels and produces a list of regions.

The extraction of geons operates as a multi-level special filter on the list of regions with each filter aiming at discovering a type of geon.

2.2.2 Arc-Geon Detection

Ellipse and circle are the two types in the arc-geon group. The arc-geon detection extracts all arc-geons from the list of regions produced by the segmentation (Let us denote by Γ that region list). Some of the regions obtained from the segmentation may already have their shapes close to an arc-geon. We first retrieve them following some heuristics before applying an ellipse detection algorithm to the remaining regions.

Detecting Arc-Geons Using Simple Heuristic Rules: Some regions in Γ may have shapes close to an arc-geon. To reduce the complexity of future steps, we try to detect as many of those regions as possible. The rules are based on the observation that given a region obtained from the segmentation phase, if the region is an ellipse or a circle region, its center abscissa and ordinate should respectively be equal to the average abscissa and the average ordinate of all points that lie in the region; similarly its major and minor axes should respectively be the longest and shortest diameters of it respectively. From the observation we can get a clear idea of the location the arc-geon to be found if any. So we construct an arc-geon from the observation parameter and try to see if the shape of the arc-geon constructed is close to a contour in the region.

Detecting Arc-Geons Using Ellipse Detection Algorithm: The previous step although fast cannot handle irregular shapes caused by the overlapping or the physical connections between two or more geons. To extract all other arc-geons, we apply an ellipse detection algorithm on the remaining regions. A number of methods for ellipse detection exist, and we have chosen the ellipse detection approach introduced by Xie and Ji in[10] for efficiency.

2.2.3 Polygon-Geon Detection

Under the assumption that all arc-geons have been extracted in the arc-geon detection procedure, we can conclude that the remaining detectable 2D geons (if any) are polygon-geons with at most four edges. Similar to the arc-geon detection, we first try to remove as many regions by detecting rectangle geons using simple heuristic rules; then we apply our Extract Geon From Line Pieces(EGFL) to detect all remaining polygon-geons.

Detecting Rectangle Geons Using Simple Heuristic Rules: Some regions in Γ may have shapes that looks like a polygon-geon. Note that, polygon-geons other than rectangle have too many flexibilities in their shapes that makes them more difficult to identify. Rectangle geons corresponding to parts of real life objects normally have one of their edges parallel to the ground (X-axis). A car body, a cabinet, a building are a few examples. For a rectangle region with one edge parallel to the ground, its lower left corner vertex should have the smallest abscissa and ordinate among all points that lie in it, and its upper right corner vertex should have the largest abscissa and ordinate among all points that lie in it. From

the observation, we create a rectangle and try to see if there is a contour that is close to the rectangle we create.

Detecting Polygon-Geons Using EGFL: We first extract straight line pieces from each remaining region in Γ, and then create all possible geons from different spatial combinations of the detected line pieces.

We analyse the possible spatial relationships between two line pieces and listed all possible eight cases as shown in Fig.2. EGFL acts differently according to the spatial relationship between two given line pieces.

Fig.2(a), 2(b), and 2(c) describe situations where two line pieces are parallel to each other. We project both two line pieces onto the X-axis and the Y-axis. If the projection of one line locates completely inside of the other as shown in the left of Fig.2(a), we construct a trapezoid by connecting the end points of two lines as shown in the right of Fig.2(b). Otherwise if the projection of one line locates partially inside or completely outside of the other as shown in the left of Fig.2(b) and Fig.2(c) respectively, we construct a parallelogram and a triangle if two line pieces differentiate in length as shown in the right of Fig.2(b) and Fig.2(c).

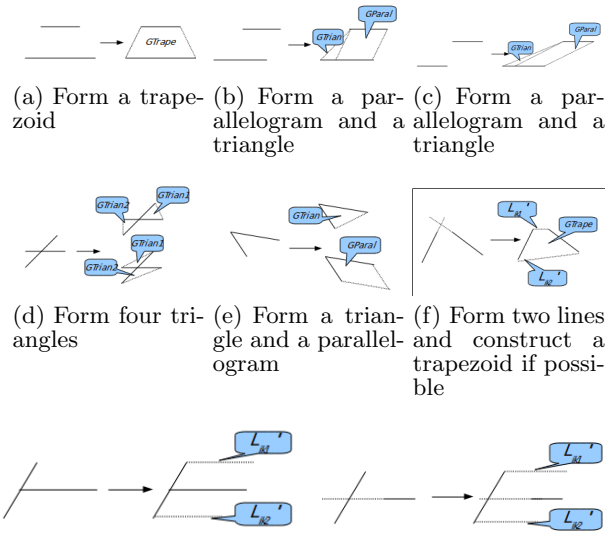
Fig.2(d) to Fig.2(h) describe all other situations. In Fig.2(d), the two line pieces cross each other. In this case, we create four triangles since all of them are possible geons. In Fig.2(e), the two line pieces have one identical end point. They can be two edges of a triangle or a parallelogram; so we create a triangle and a parallelogram. Fig.2(f) shows the situation that two line pieces will intersect on the extensions of both of them. They can come from a triangle or a trapezoid, but since the cost of substituting a triangle with a trapezoid is negligible as explained in Section3.2.1, we construct a trapezoid. In Fig.2(g), an end point of one of the line pieces (Let us denote by l_1 the line piece) also belongs to the other line piece, and in Fig.2(h), the two line pieces will intersect if we extend one of them (Let us denote by l_2 the line piece that need to be extended). None of the defined six 2D geons contains edges with spatial relationships in these two cases, however, it is possible that there exist two geons that share the line piece l_1 in the case of Fig.2(g) or the line piece l_2 in the case of Fig.2(h). In this case, we create two more line pieces which are parallel to l_1 or l_2 as shown in the right of Fig.2(g) or Fig.2(h) respectively and add them into the line piece list.

For any polygon-geons, EGFL needs only two of its edges to extract it. Consequently, even for a geon of irregular shape, EGFL is capable of detecting it if any two of its composing edges have been extracted.

A geon extracted from a region should have most of its pixels within the region. For each geon, let us denote by n_g the number of pixels that lie in both the geon and the region that the geon extracted from, and let us denote by a_g the area of the geon. To validate a geon, we compute the ratio between its n_g and a_g and check whether the result is larger than some predefined threshold.

3. MATCHING GEON SEQUENCE AND OBJECT MODELS

This section discusses the method we used to map a geon list to given models. We represent the geon list with a 2D string [6] with each character representing a geon. The order of characters is determined based on the geometrical positions of the geons. The object models are also converted



(a) Form a trapezoid (b) Form a parallelogram and a triangle (c) Form a parallelogram and a triangle

(d) Form four triangles (e) Form a triangle and a parallelogram (f) Form two lines and construct a trapezoid if possible

(g) Form two lines and add them into line list (h) Form two lines and add them into line list

Figure 2: Geon Construction From Two Line Pieces

to a 2D string. Then, we compute the weighted levenshtein distance between the two 2D strings.

3.1 Generating Geon 2D String

We define a geon 2D string as two lists of geons sorted in ascending order of their center ordinates firstly and then their center's abscissas as in [6]. The raw geon list extracted is unsorted, and it does not contain any position relationship information. Representing geons in a 2D string ensures that the spatial relationships among the geons is taken into account.

3.2 Distance Function

We need to calculate the distances between the geon 2D string and a model 2D string first. We then categorize the image based on the distances. We calculate the distance using an adapted levenshtein distance algorithm(ALDA).

3.2.1 Adapted Levenshtein Distance Algorithm

For a given geon 2D string a model 2D string, we compute the distance between them using an adapted levenshtein distance algorithm. The levenshtein distance, also called edit distance, measures the level of difference between two sequences. The original levenshtein distance for two strings represents the minimum number of insertion, deletion, or substitution operations to transfer one string to another, and operation weights for all those three operations are the same.

Assigning the same weight to all operations for our case won't be appropriate. 2D geons are extracted in an approximate way, which means, a rectangle geon could be extracted as an ellipse or a trapezoid. One main reason for this phenomena is that a target object might get blocked by other objects in an image; another reason is the segmentation algorithm cannot always be trusted; and also, different view points of an object can produce different types of geons. So we need to assign different weights to different operations. For example, the cost of substituting a rectangle to a parallelogram should be less than the cost of substituting a rectangle to a triangle since rectangle is more similar to

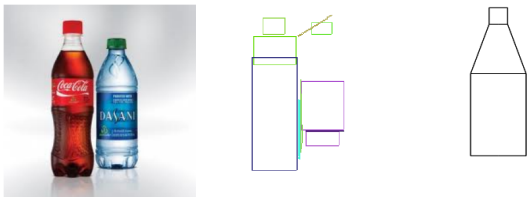
Table 1: Experiment results on CalTech dataset

Type	Size	2DPCA(%)		OurApproach(%)	
		precision	recall	precision	recall
Car-Rear	30	**75.7778**	91.1111	59.5385	**100.000**
	45	**70.9036**	90.0000	61.5385	**100.000**
Leaf	30	**86.6667**	**78.6667**	44.4350	57.1429
	45	**85.9206**	**76.6667**	40.4353	70.0000
Air Plane	30	**54.3095**	52.0000	44.6675	**80.0000**
	45	**80.0317**	62.8571	53.8461	**86.6667**

parallelogram than to triangle. We defined different weights for different operations in our ALDA.

3.2.2 *Generating Sub Geon Strings*

An extracted geon list might contain noisy geons as shown in Fig.3. Fig.3(a) is the input image, Fig.3(b) is the extracted geon list, and in Fig.3(c) is a model of bottle. As discussed in 3.2.1, when computing the distance, if we need to delete some geon from the geon string, the deletion operation cost will be added into the distan



(a) Input Image (b) Raw Geons (c) Bottle Model

Figure 3: Matching Failure Caused by Noises

The presence of noisy geons increases the distances inappropriately. To address this problem, we compute the distances between all combinations(a n choose i problem) of an extracted geon string and a model, and select the minimum distance as the final result. When generating sub geon strings, we also make sure not to remove a geon which is part of an object by restricting that geons being removed can only be outliers.

After we have calculated the distances between a geon string and all given models, we can recognize the object as the type of the model which gives the smallest distance.

4. EXPERIMENT RESULTS

We compared our matching method with the 2DPCA(2 Dimensional Principle Component Analysis)[11]. Compared to other feature-based object recognition competitors, our approach does not need any training data, but it does require pre-defined object models. We have constructed an image dataset *CSID*(Complex Small Image Dataset). Images in *CSID* have relative more complex scenes, smaller number of images when compared to modern image datasets, such as calTech, COIL-100[8]. There are currently 3 types of objects in *CSID*: car, lamp, and bottle, each type contains 20 images. Experiments have been carried on the *CSID* and COIL-100.

Table.1 shows the experimental results based on COIL-100 and Table.2 shows the experimental results based on *CSID*. In general, we can find that 2DPCA-SVM has better performance on **precision** than our approach on COIL-100 dataset, and our approach has better performance on **recall** for **car-rear** and **airplane**. We also discovered that the 2DPCA's performance drops when the size of dataset drops.

Table 2: Experiment Result on CSIDataSet

Object Type	Our Approach(%)		2DPCA(%)	
	Recall	Precision	Recall	Precision
Bottle	**100.00**	**58.323**	33.000	25.000
Car	**100.00**	58.822	50.000	**66.667**
Lamp	**100.00**	44.383	50.000	**75.000**

Our approach is less performant on the **leaves** dataset compared to the 2DPCA. On one hand, all images in the **leaves** dataset have been taken from the same or very similar leaf object in different backgrounds. On the other hand, it is very hard for our approach to decompose leaves into geons, and also hard to design proper models for leaves.

Table.2 shows that 2DPCA has poor performance on our **CSID** and our approach almost obtained the same performance as for the calTech dataset shown in table.1. Our approach gives perfect **recall** values for all classes, which means our approach can retrieve all images of a certain class from the dataset.

5. CONCLUSION AND FUTURE WORK

In this paper, we first defined a novel parametric 2D geon set based on existed 3D geon set. Then we presented a scheme to extract 2D geons from 2D images. Experiment results showed that the scheme worked for not only synthetic images, but also for complex real world images in most cases. However, our approach is restricted by the segmentation result of images. In the future, we will focus on finding a way to replace the segmentation part and obtaining more stable results.

6. REFERENCES

[1] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(5):115–147, November 1987.

[2] L. Du and R. Munck-Fairwood. Geon recognition through robust feature grouping. *Image Processing, 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden*, June 1995.

[3] R. Fairwood. Recognition of generic components using logic-program relations of image contours. *Image and Vision Computing*, 9(2):113–122, April 1999.

[4] P. F. Fekzebszwakv. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[5] P. J. Lazebnik S., Schmid C. Beyond bags of features: Spatial pyramid matching for natural scene catergories. *CVPR*, pages 2169–2178, 2006.

[6] A. J. Lee and H. Chiu. 2d z-string: A new spatial knowledge representation of image databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, Jan 2004.

[7] Y. Lu and H. Guo. Background removal in image indexing and retrieval. *Proceedings 10th Int.Conf, Image and Processing, IEEE Computer Society*, pages 933–938, 1994.

[8] M. H. Nene S.A., Nayar S.K. Coil 100 database. 1996.

[9] K. Wu and M. D. Levin. 3-d shape approximation using parametric geons. *Image and Vision Computing archive*, 15(12), Jan 1996.

[10] Y. Xie and Q. Ji. A new efficient ellipse detection method. *International Conference on Pattern Recognition 2002*, (2):859–862, 2004.

[11] J. Yang, D. Zhang, A. F.Frangi, and J. Yang. Two dimensional pca: A new approach to appearance based face reprensentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, Jan 2004.