A New Object Detection Technique Based on Geometric Hashing

Ling Ma
2791-5 Fast Corporation,
Shimoturuma,Yamato,
Kanagawa,Japan
L.Ma@fast-corp.co.jp

K. Masuda, J. Sugano, A. Yamaguchi
2791-5 Fast Corporation
Shimoturuma,Yamato,
Kanagawa,Japan
Kt.Masuda@fast-corp.co.jp

Abstract

This paper describes the modifications of Geometric Hashing pattern recognition scheme to fulfill the requirements of real-time industrial applications in recognizing general shapes with large numbers of feature points. The scheme is robust to variety of object transformations and partial occlusion. By decomposing the pattern into sub-patterns, using a plurality of hash tables to record the sub-patterns and by using Image Pyramid in combination with edge direction information, the modeling and recognition performance was improved.

Keywords: object recognition, Geometric Hashing, invariant matching, edge detection, real-time computer vision.

1 Introduction

The recognition stage is the ultimate goal of many vision systems. For example, moving around and safely avoiding objects, picking up and placeing various objects, inspecting objects, and performing many other tasks. Model-based object recognition methods have demonstrated good performance for a variety of problems. These methods are robust to partial occlusion as well as clutter. Generalized Hough Transformation (GHT) [1] and Geometric Hashing [2]-[4] are two model-based methods. Normally, GHT requires much memory and long calculation time. Geometric Hashing is a general method that can recognize objects that may have undergone various transformations such as similarity, affine and projective transformations. This method seems to be very effective when the object to be recognized has a shape that can be expressed by a few feature points. With objects having a general shape, however, too many feature points may often be obtained, in which case the method is unfavorable in recognizing speed. This paper shows how the evidence of using image edges as feature points can be decomposed into sub-patterns, and how to use the plurality of hash tables to record sub-patterns, as well as pyramid techniques in combination with edge direction information to speed up modeling and recognition phases. Experimental results show that the proposed technique is effective and efficient for shape recognition with a large number of feature points

After giving a short introduction to the basic principle of geometric hashing in Section 2, we will present the modifications of Geometric Hashing method in Section 3, which improves the recognition efficiency. Finally we will show the results for verifying the effectiveness of the proposed method and the conclusion.

2 Shape Matching by Geometric Hashing

Geometric Hashing is a general model-based pattern matching technique, which is robust to partial occlusion as well as clutter. With Geometric Hashing, an object to be recognized is represented by a group of feature points and consists of two phases: modeling and recognition. In this paper, we will consider the similarity transformation. The main goal is to represent the feature points by a few of the intrinsic parameters in a rotation, scale and translation invariant manner. This is done by defining an orthogonal coordinate frame based on the ordered pair of points (referred to as basis-pair) from the point set and representing all other feature points with their coordinates in this local frame. Such a coordinate frame is defined uniquely by assigning coordinates (-1/2,0) and(1/2,0) to the first and second points respectively. Any similarity transformation applied to the point set will preserve their coordinates, if they are represented in the same two-point basis. So, if the two-point basis selected in the recognition phase is in the same order as in the modeling phase, the coordinates of the other points in the two local frames will be the same even if the object has undergone similarity transformations. In order to recognize partial occluded objects, various basis-pairs can be selected, so even if some feature points were occluded in the recognition phase, other basis-pairs can be used to compare the object with the model pattern. For recording the local frames, a hash table is used. For every ordered pair of model feature points, the coordinates of other model feature points are computed in the orthogonal local frame defined by this ordered pair. Each coordinate like this (after a proper quantization) is used as an entry to a hash table, where the basis-pair at which the coordinate was obtained is recorded.

The models are assumed to be known in advance, and thus allow preprocessing, as follows:

1 Extract pattern feature points.
2. Pick one reference frame by choosing two ordered feature points.
3. Compute the orthogonal basis associated with this reference frame.
4. Compute the coordinates of all other feature points in this reference frame.
5. Use each coordinate as an address to the hash table. Store the entry at the hash table address.
6. Repeat step 2 to 5 for each pattern reference frame.

The recognition stage of the algorithm uses the hash table, prepared in the preprocessing step. The matching of a target object is accomplished as follows:

1. Extract feature points in the scene.
2. For each reference frame of the target.
3. Compute orthogonal basis associated with this reference frame.
4. Compute the coordinates of all other points in the current reference frame.
5. Use each coordinate to access the hash-table and retrieve all the records.
6. “Votes” for the recorded pairs.
7. Compute the transformations of the ``high scoring" hypotheses.
8. Repeat step 2 to 7 for each target reference frame.

If the number of model feature points is m, the modeling complexity will be $O(m^3)$. However, if the number of the feature points in the scene is ***n***, the recognition complexity will be $O(n^3)$.

3 Performance Improvement of Geometric Hashing

The complexities of the modeling and recognition phases are in the third order of the pattern feature points and scene feature points respectively. If there are a large number of feature points in the pattern and the scene, the modeling and recognition processes will be time consuming and unfavorable for industrial applications. Though the modeling phase can be completed off -line, it is still expected that the process should be as fast as possible. If we use only one hash table to record each pattern that has a large number of feature points, the number of all possible basis-pairs will be large and there will be a large number of local orthogonal frames, in each of which a large number of local coordinates have to be computed. Also, some bin's list will be very long and much time will be consumed in accessing it in the modeling and recognition phases. This section presents some techniques to improve the performance of the Geometric Hashing methods so that it can be used in real-time recognition applications.

3.1 Pattern Clustering and a plurality of hash tables

The possible method to speed up the modeling and recognition phases is to reduce the feature points in the pattern and in the scene. A small number of feature points means that the number of possible basis-pairs and the computation of the local coordinates of all feature points are small and this helps to shorten computation time. Obtaining a small number of feature points from free-form shape objects is not easy task. Accordingly, we propose two methods for reducing the number of basis's and the amount of time needed to coordinate calculations for feature points, without reducing the number of feature points.

First we suggest decomposing a pattern into sub-patterns by using the clustering method or other segment methods. In this way, for every sub-pattern, we only need to select basis pairs on the same sub-pattern, and to compute local frame coordinates of the feature points that sub-pattern. This greatly reduces the number of possible basis-pairs, and the computation time of local frame coordinates.

Second, we suggest registering different sub-patterns in different hash tables. This reduces the length of the basis-pairs list in the hash table bins and the time needed to access the memory.

If the number of the original pattern's feature points is ***m*** and we decompose the pattern into ***k*** sub-patterns with the average number of ***n*** feature points, then a plurality of sub-patterns will make the computation $1/k^2$ of that one pattern. For every sub-pattern, we only select a basis-pair of this sub-pattern and compute if local frame coordinates of the other feature points only for this sub-pattern, we can use the coordinates as an entry to the corresponding hash table.

In the recognition phase, we can also use the sub-pattern's information to limit the range, within which the basis-pair is selected and the scene local frame coordinates are computed. Feature points paired by distance larger than the maximum diameter of the sub-patterns, will not be selected as basis-pair to vote any sub-pattern. This reduces the number of possible basis-pairs in the recognition phase. For every selected basis-pair, the local frame coordinates of the feature points outside a certain range do not need to be computed to vote since they will not match any sub-pattern. This reduces the number of the local frame coordinates to be computed and the time of memory access; hence it speeds up the recognition phase.

For basis-pairs (O_1, O_2) in a scene, if the pattern basis-pair (P_1, P_2) got a large number of votes, we can get one object pose (x, y, θ, s) by comparing the length, angle and location of the two line segments.

After getting all the possible poses, we can define a metric in a posed space and use the clustering method to get the object pose(s). We can also check the pose(s) by their edge points and refine the pose(s) by the least square method.

The method to compute local frame coordinates is as follows: given one basis-pair (P_i, P_j), the local frame is:

$$X_{ij} = (P_j - P_o) \Big/ \left\| P_j - P_o \right\| = (x_{ij}, y_{ij}) \quad ;$$

$$Y_{ij} = -(y_{ij}, x_{ij}) \, ; \quad P_o = (P_i + P_j) \big/ 2$$

The coordinates of any other feature point P_f in this frame are:

$$(x_f, y_f) = (X_{ij} \bullet (P_f - P_o), Y_{ij} \bullet (P_f - P_o))$$

The coordinates are used to record the basis-pairs in the hash tables in the modeling phase and to access the hash tables to vote the pattern basis-pair in the corresponding bins in the recognition phase.

3.2 Image Pyramid, Edge direction and basis pair selection

In order to improve the performance, we can also use other techniques to reduce the number of edge points and the number of basis-pairs. One of the methods to reduce the number of edge points is the Image Pyramid. Also, we can select basis-pairs by limiting the angle between the normal directions at the two feature points: if the angle is not in some range, they will not be selected as a basis-pair.

To summarize, the modeling and the recognition procedures are as follows:

Modeling phase:

① Filtering, edge extracting, and calculating edge directions, sampling edge by pyramid technique.

② Decompose the pattern edge into sub-patterns.

③ For each ordered pair of edge point on The same sub-pattern, if the angle between the edge directions of the pair is in a certain range, it can:

a) Compute the local frame and the local frame coordinates of the other edge points on the same sub-pattern.

b) After a proper quantization, use the coordinates as an index to the corresponding hash table and insert into the hash table bin the information of the basis-pair, which was used to determine the coordinates.

Recognition phase:

① Detect the viewed image edge, sampling edge by using the Pyramid Technique.

② Choose an ordered pair of edge point in the image, if the angle between the edge directions of the two edge points and the distance between them are in a certain range, it can:

a) Compute the local frame and the local frame coordinates of other edge points, which are in a certain close range of the basis-pair.

b) Appropriately quantize each such coordinate and access bins of all hash tables, then for every entry there cast a vote to the pattern basis-pair.

c) Determine the pattern basis-pair that maximum the scores of all pattern basis-pairs in the plurality of the hash tables. If it scores high enough, calculate the pose.

③ Find the appropriate poses by clustering the poses. Check and refine the poses using the basis patterns and scene image edge points.

4 Results and conclusion

4.1 Results

The images below demonstrate the efficiency and the effectiveness of the method in this paper. The size of all images is 512×480.The test environment is 2.8GHz CPU with 512M memory and VC++6.0

Figure 1 shows a sample of finding objects that have been scaled, rotated and partial occlusion. The search time was about 60 ms, including edge detection, vote, pose clustering and posed refining with vote time less than 10 ms. This sample uses one pattern and one hash table.

Figure 2 shows a character string pattern detection sample. The total search time was about 90 ms with vote time less than 12 ms. This sample uses 4 sub-patterns to represent two characters and 4 hash tables.

Figure 3 shows a color pattern search sample. The objects have a different size, different angles and different colors. After detecting the edges in the color image, we can detect objects in the scene by the presented method. The total search time was about 65 ms with vote time less than 2 ms.

In these samples we used FVL (Vision software Library developed by FAST CORPORATION) in image preprocessing to remove clutter and to detect the image edge.

4.2 Conclusion

In this paper we described the modifications of Geometric Hashing pattern-matching methods to fulfill the requirements of real-time industrial applications. Since Geometric Hashing is model-based, it is robust against partial occlusion and clutter. We demonstrated that the efficiency can be improved by decomposing pattern into sub-patterns and recording the sub-patterns into a plurality of hash tables using the Pyramid Technique, in combination with edge direction information; to reduce the number of edge points and the number of local frames in the modeling and recognition phases.

Acknowledgements:

Thanks to Ms. Mami Makino for her English translation of this paper.

References

(1) D.H.Ballard: Generalizing the Hough transformation to detect arbitrary shapes. Pattern Recognition. Vol.13【2】.pp111－122. 1981

(2) Lamdan,Y., and H.Wolfson. Geometric Hashing: A general and efficient model-based recognition scheme: In Proceedings of International conference on Computer vision.pp238-249.1988

(3) Lamdan,Y., J.Schwartz and H.Wolfson: Affine invariant model-based object recognition: IEEE Transactions on Robotics and Automation.6(5) pp578-589,1990

(4) H.Wolfson and I. Rigoutsos. Geometric Hashing: an overview. IEEE Computational Science and Engineering, 4(4):10–21, Oct-Dec 1997.

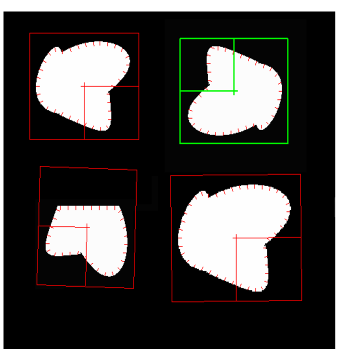

Figure 1: scaled, rotated, occluded object detection

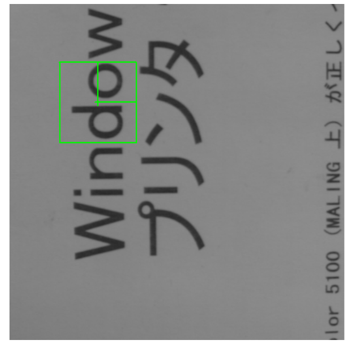

Figure 2: Character string pattern detection

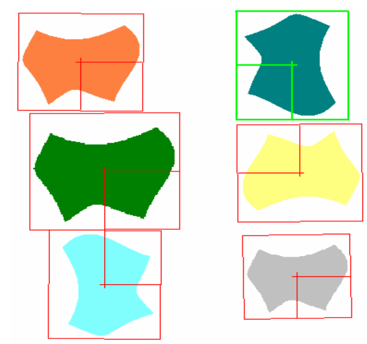

Figure 3: Color object detection

About the author: Ling Ma Graduated from Beijing University of Aeronautics & Astronautics with a PhD degree in Department of Manufacturing Engineering in 1997. He currently works for FAST Corporation, which is an Image Processing Technology Company in Japan and specializes in image processing, pattern recognition, 3D machine vision and computer graphics.