

# Feature Collation based on The Generalized Hough Transform

Lijuan Song<sup>1,2,\*</sup>

1. School of Information Science and Technology, Northwest University, Xi'an 710069, P.R.China;

2. School of Mathematics and Computer Science, Ningxia University, Yinchuan, China 750021, P. R. China

\* slj@nxu.edu.cn

**Keywords:** feature collation; maximal clique; generalized Hough transform

**Abstract:** This paper mainly discussed the problem of recognizing objects by their features. The maximal clique approach is seen to be capable of finding solutions to recognize objects and scene analysis. The main focus of this thesis is studied that the GHT is particularly well suited to object detection in real space as it is one type of spatial matched filter. Finally, this paper compared the computational requirements of the maximal clique and GHT approaches to object location. The computational problems of the GHT are examined more generally.

## Introduction

The generalized Hough transform is of great importance for the detection of features such as lines, circles, and ellipses, and for finding relevant image parameters. And the GHT can detect arbitrary shapes[1].

Gope and Kehtarnavaz(2007) have demonstrated a new method for affine matching between planar point sets[2]. The method makes use of the convex hulls of the point sets and performs matching between them: this is a useful approach because (a) convexity is affine invariant and (b) use of the convex hull is intrinsically robust. Silletti(2011) have devised a variant approach to spectral graph matching in which new similarity measures are applied[3]. The approach permits application to variety of types of image and yields results that are said to show significant improvements over certain preexisting methods. When using the GHT to locate an object, edge pixels are detected and used to compute candidate positions for the localization point L. To achieve this, it is necessary to threshold the edge gradient magnitude[4]. How should the threshold be chosen?

## The generalized Hough transform

The standard Hough technique can detect arbitrary shapes. In principle, it is trivial to achieve this. First, we need to select a localization point L within a template of the idealized shape. Then, we need to arrange such that, instead of moving from an edge point a fixed distance R directly along the local edge normal to arrive at the center, as for circles, we move an appropriate variable distance R in a variable direction  $\varphi$  so as to arrive at L: R and  $\varphi$  are now functions of the local edge normal direction  $\theta$ . There is shown in Fig.1.

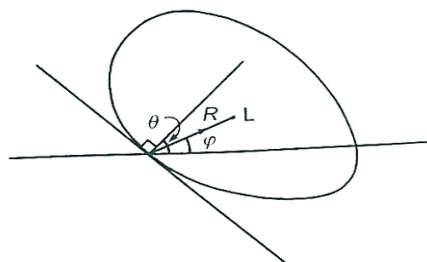


Fig.1. Computation of the GHT

Under these circumstances, votes will peak at the preselected object localization point  $L$ . The functions  $R(\theta)$  and  $\varphi(\theta)$  can be stored analytically in the computer algorithm, or for completely arbitrary shapes they may be stored as lookup tables. In either case, the scheme is beautifully simple in principle but two complications arise in practice. The first arises because some shapes have feature such as concavities and holes, so several values of  $R$  and  $\varphi$  are required for certain values of  $\theta$  as shown in Fig.2. The second arises because we are going from an isotropic shape to an anisotropic shape, which may be in a completely arbitrary orientation.

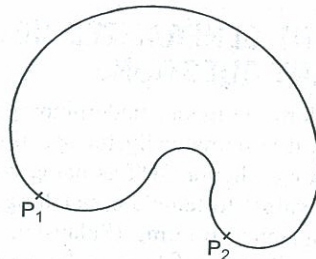


Fig.2. A shape exhibiting a concavity

To cope with the first of these complications, the lookup table must contain a list of the positions  $r$ , relative to  $L$ , of all points on the boundary of the object for each possible value of edge orientation  $\theta$ . Then, on encountering an edge fragment in the image whose orientation is  $\theta$ , estimates of the position of  $L$  may be obtained by moving a distance  $R=r$  from the given edge fragment. Clearly, if the  $R$ -table has multivalued entries, only one of these entries can give a correct estimate of the position of  $L$ . However, at least the method is guaranteed to give optimum sensitivity, since all relevant edge fragments contribute to the peak at  $L$  in parameter space. This property of optimal sensitivity reflects the fact that the GHT is a form of spatial matched filter: it is analyzed in more detail below.

The second complication arises because any shape other than circle is anisotropic. Since in most applications (including industrial applications such as automated assembly) object orientations are initially unknown, the algorithm has to obtain its own information on object orientation. This means adding an extra dimension in parameter space (Ballard, 1981). Then each edge point contributes a vote in each plane in parameter space at a position given by that expected for an object of given shape and orientation. Finally, the whole of parameter space is searched for peaks, the highest points indicating both the locations of objects and their orientations.

We may explore the theory underpinning the GHT. There are three problems need to be addressed:

1. The parameter space weighting problem: In introducing the GHT, Ballard mentioned the possibility of weighting points in parameter space according to the magnitudes of the intensity gradients at the various edge pixels. But when should gradient weighting be used in preference to uniform weighting?
2. The threshold selection problem: When using the GHT to locate an object, edge pixels are detected and used to compute candidate positions for the localization point  $L$ . To achieve this, it is necessary to threshold the edge gradient magnitude. How should the threshold be chosen?
3. The sensitivity problem: Optimum sensitivity in detecting objects does not automatically provide optimum sensitivity in locating objects, and vice versa. How should the GHT be optimized for these two criteria?

### The maximal clique algorithm

We give the maximal clique algorithm. The checking of which subgraphs are maximal cliques is a simple problem. However, in real matching tasks it can quickly become unmanageable.

There is shown in Table 1. It shows that is perhaps the most obvious type of algorithm for finding maximal cliques. It operates by examining in turn all cliques of a given number of nodes

and finding what cliques can be constructed from them by adding additional nodes. This permits all cliques in the match graph to be identified. However, an additional step is needed to eliminate all cliques that are included as subgraphs of a new larger clique before it is known which cliques are maximal.

Table 1. The maximal clique algorithm.

---

```

set clique size to 2; //this is the size already included by the match graph
While (newcliques=true) { //newcliques still being found
  increment clique size;
  set newcliques=false;
  for all cliques of previous size
    { set all cliques of previous size to status maxclique;
      for all possible extra nodes
        if extra node is joined to all existing nodes in clique
          { store as a clique of current size;
            set newcliques=true;
          }
    }
  // the larger cliques have now been found
  for all cliques of current size
    for all cliques of previous size
      if all nodes of smaller cliques are included in current clique
        set smaller clique to status not maxclique;
  // the subcliques have now been relabeled
}

```

---

In view of the evident importance of finding maximal cliques, many algorithms have been devised for the purpose. It is probable that the best of these is now close to the fastest possible speed of operation. Unfortunately, the optimum execution time is known to be bounded not by a polynomial in  $M$  but by a much faster varying function. Specifically, the task of finding maximal cliques is akin to the well-known traveling salesman problem and is known to be “NP-complete” implying that it runs in exponential time. Thus, whatever the run-time if for values of  $M$  up to about 6, it will typically be 100 times slower for values of  $M$  up to about 10, and 100 times slower again for  $M$  greater than  $\sim 14$ . In practical situations, we can use the GHT to tack this problem.

### Using the GHT for feature collation

How the GHT can be used as an alternative to the maximal clique approach, to collate information from point features in order to find objects. To apply the GHT, we first list all features and then accumulate votes in parameter space at every possible position of a localization point  $L$  consistent with each pair of features as shown in Fig.3. This strategy is particularly suitable in the present context, as it corresponds to the pairwise assignments used in the maximal clique method. To proceed, it is necessary merely to use the interfeature distance as a lookup parameter in the GHT R-table. For indistinguishable point features, this means that there must be two entries for the position of  $L$  for each value of the interfeature distance. Note that we have assumed that no symmetries exist and that all pairs of features have different interfeature distances. If this were not so, then more than two vectors would have to be stored in the R-table per interfeature distance value.

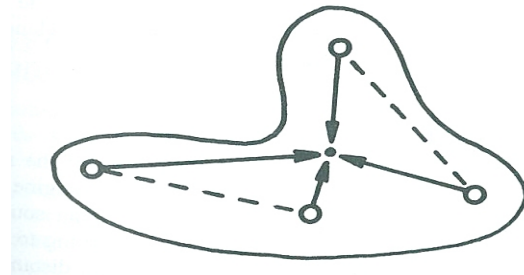


Fig.3. Method for locating L from pairs of feature positions

In Fig.3, each pair of feature points gives two possible voting positions in parameter space, when objects have no symmetries. When symmetries are present, certain pairs of features may give rise to up to four voting positions.

To illustrate the GHT approach outlined above, we apply it to the triangle example shown in Fig.4. Fig.4(c) shows the positions at which votes are accumulated in parameter space. There are four peaks with heights of 3,1,1 and 1, it being clear that, in the absence of complicating occlusions and defects, the object is locatable at the peak of maximum size.

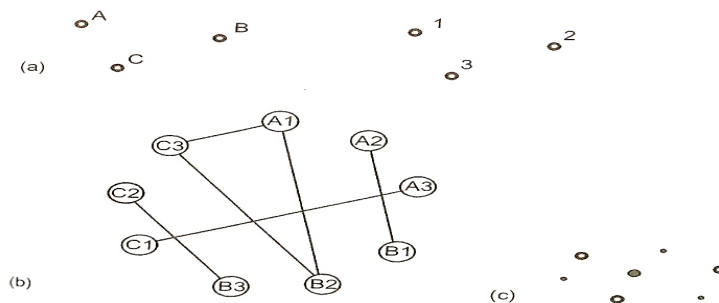


Fig.4. A matching problem-a general triangle

We compare the computational requirements of the maximal clique and GHT approaches to object location.

For an object possessing  $n$  features, the match graph contains  $n^2$  nodes, and there are  $n^2 C_2 = n^2(n^2-1)/2$  possible pairwise compatibilities to be checked in building the graph. The amount of computation at this stage of the analysis is  $O(n^4)$ . To this must be added the cost of finding the maximal cliques. Since the problem is NP-complete, the load rises at a rate that is faster than that of polynomial, and probably exponential in  $n^2$ .

Now consider the cost of getting the GHT to find objects via pairwise compatibilities. As has been seen, the total height of all the peaks in parameter space is in general equal to the number of pairwise compatibilities in the match graph. Hence, the computational load is of the same order,  $O(n^4)$ . Next comes the problem of locating all the peaks in parameter space. In this case, parameter space is congruent to image space. Hence, for an  $N \times N$  image only  $N^2$  points have to be visited in parameter space and the computational load is  $O(N^2)$ . However, that an alternative strategy is available in which a running record is kept of the relatively small numbers of voting positions in parameter space. The computational load for this strategy will be  $O(n^4)$ : although of a higher order, this often represents less computation in practice.

## Conclusion

The task of recognizing objects by their features tends to involve considerable computation and GHT can in some cases provide a satisfactory solution to these problems. When this happens, it is because the graph theory representation is not well matched to the relevant real-space template matching task in the way that the GHT is. The GHT is particularly well suited to object detection in real space as it is one type of spatial matched filter.

## Acknowledgment

The paper is supported by Ningxia University Research Projects (NGY2014055). I would like to thank the other members: Prof.Peng and Prof.Wang, for their insightful comments and suggestions.

## References

- [1] Kasif,s.,Kitchen,L.,Rosenfeld,A.. A Hough transform technique for subgraph isomorphism.Pattern Recogn.Lett.2,83-88,1983.
- [2] Gope,C.,Kehtarnavaz,N.. Affine invariant comparison of point-sets using convex hulls and Hausdorff distances. Pattern Recogn. 40,309-320,2007.
- [3] Silletti,A., Abate,A.,Axelrod,J.D.,Tomlin,C.J..Versatile spectral methods for point set matching. Pattern Recogn.Lett.32(5),731-739,2011.
- [4] Aguilar,W.,Frauel,Y.,Escolano,F.,Martinez-Perez,M.E.,Espinosa-Romero,A..A robust graph transformation matching for non-rigid registration. Image Vis. Comput. 27,897-910,2009.