Geometric Hashing

Yansheng Ming
Hash table and hash function
Geometric object recognition
Geometric object recognition

Is there a transformed (rotated, translated and scaled) subset of some model point-set which matches a subset of the scene point-set?
Geometric object recognition

The difficulties of matching point sets:
• Unknown transformation
• Outliers
• Noise
• Large number of models

The naïve approach results in worst case $n^5$ complicity for recognition of a single model (similarity transformation). Geometric hashing reduce it to $n^3$. 
Preprocessing phase
Preprocessing phase
Preprocessing phase

For each model $m$ do the following:

1. Extract the model’s point features. Assume that $n$ such features are found.

2. For each ordered pair, or basis, of point features do the following:
   a) Compute the coordinates $(u,v)$ of the remaining features in the coordinate frame defined by the features.
   b) After proper quantization, use the tuple $(u_q, v_q)$ as an index into 2D hash table data structure and insert in the corresponding hash table bin the information $(m, (basis))$, namely the model number and the basis tuple used to determine $(u_q, v_q)$.
Recognition phase

Model

Image

Cast 1 vote for each entry in bin's list

In the end histogram all entries with one or more votes
Recognition phase

1. Extract the various points of interest.
2. Choose an arbitrary ordered pair of the points as the basis.
3. Computer the coordinates of the remaining interest points.
4. Quantize each such coordinate and access the hash table bin; for every entry found there, cast a vote for that \((m, (basis))\).
5. Histogram all hash table entries. Find entries received votes exceeding a threshold. Each entry is a potential match.
6. For each potential match, recover the transformation \(T\) that results in the best least-squares match between all corresponding feature pairs.
7. Verify the matching under \(T\). If succeed, end the program; otherwise, go to step 2.
Application 1

Recognition of 3D objects from 2D images

Figure 3: a) a gray scale image of a 'short car'; b) a gray scale image of two overlapping 'cars'; c) the scene after the 'low-level' processing; d) recognition of the 'cars'.
Application 2

Recognition of polyhedral objects
Summary

• Recognize objects undergoing arbitrary transformation.
• Recognize partial occluded objects.
• Efficiency in time.
Reference


• WiKi: Geometric Hashing