Geometric Hashing

Feb. 27. 2007

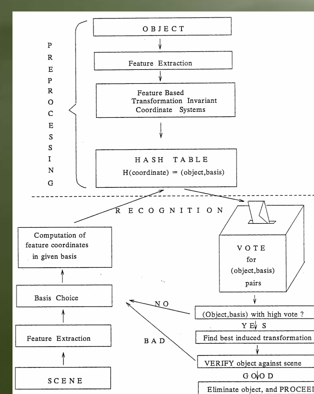
Goo Jun

Overview (1)

- Origins of Geometric Hashing
  - Developed as a solution to computer vision problem
  - Find common substructures in a scene regardless of rotation, translation, scale, and occlusion
- Applications
  - Robotic memory of a tool on an assembly line
  - Database image queries
  - Structural molecule comparison (Protein)
  - Security – fingerprint and face recognition
  - Medical – detection of irregularities on images (MRI,X-ray)

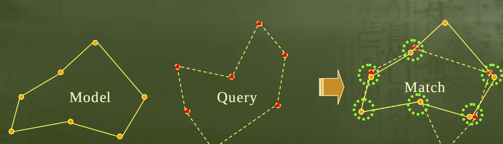
Overview (2)

- Several Aspects of Geometric Hashing
  - Ability to recognize objects that have undergo an arbitrary transformation.
  - Can perform partial matching.
  - Efficient and can be parallelized easily.
  - Use transformation-invariant access key to the hash table.
  - Two phases (preprocessing and recognition).
  - Require a large memory to store hash table.



Recognition under Similarity Transformations

- Problem Definition:
  - "Is there a transformed (rotated, translated, and scaled) subset of some model point-set which matches a subset of the scene point-set." (Lamdan, Wolfson, 1988)

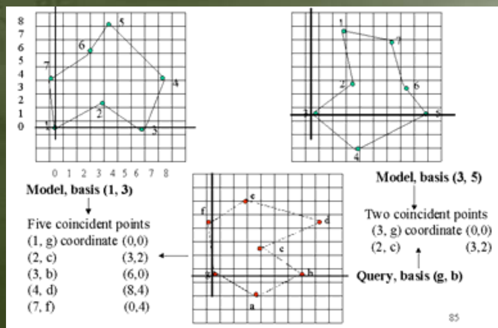


Brute-force search will be O(n^2 * m^2 * k)

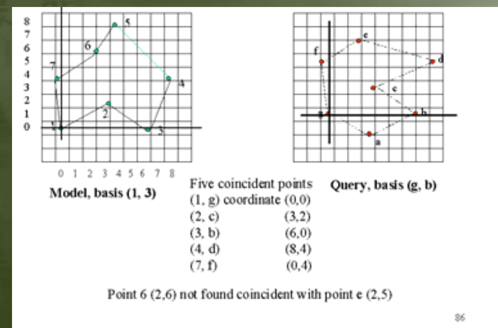
Reference Frames

- A method to compare two sets of points in similarity-transform invariant way
  - Define coordinate systems for both figures (A, B), called reference frames
  - Two points (basis pair) can define a reference frame, e.g., origin at one of them, and one of the axis through both points.
  - The coordinates of the points are computed in the reference frame, constituting a reference frame system
  - Count how many pair of points (one from each figure) have the same coordinates

Example: Reference Frame – (1)



Example: Reference Frame – (2)



Remarks on Reference Frames

- The number of coincident points depends on the resolution of the coordinate system, and on the *basis pairs* used.
- Generally, all combination of points should be used as basis pairs, resulting in comparing (m(m-1)/2)x(n(n-1)/2) reference frame systems
- Using all combinations might introduce redundancy. Let (a_i, a_k) and (b_j, b_l) be the basis pairs, and (a_r, b_u) and (a_s, b_v) both coincide. Then it is likely that the same coincidence set is found if (a_r, a_s) and (b_u, b_v) are used as basis. *Note however that similarity and not exact equality is used.*

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 basis.
     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).

Example – Preprocessing (1)

- Construct Hash Table

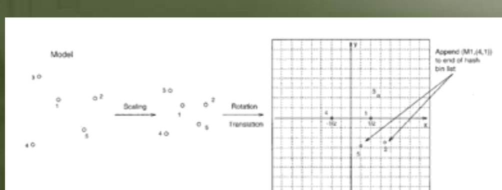


Figure 1. Determining the hash table entries when points 4 and 1 are used to define a basis. The models are allowed to undergo rotation, translation, and scaling. On the left of the figure, model *M* comprises five points.

Example – Preprocessing (2)

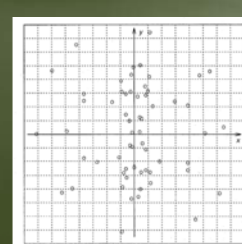
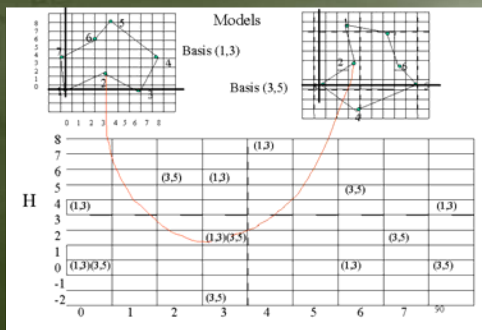


Figure 2. The locations of the hash table entries for model M. Each entry is labeled with the information "model M" and the basis pair (i, j) used to generate the entry. The models are allowed to undergo rotation, translation, and scaling.

Example – Preprocessing (3)



Recognition Phase

- When presented with an input image, do the following:
  1. Extract the various points of interest. Assume that S such interest points found.
  2. Choose an arbitrary ordered pair, or basis, of interest points in the image
  3. Compute the coordinates of the remaining interest points in the coordinate system that the selected basis defines
  4. Appropriately quantize each such coordinate and access the appropriate hash table bin; for every entry found there, cast a vote for the model and the basis.

Recognition Phase – cont'd

5. Histogram all has table entries that received one or more votes during step 4. Proceed to determine those entries that received more than a certain number, or threshold, of votes. Each such entry corresponds to a potential match
6. For each potential match discovered in step 5, recover the transformation T that results in the best least-squares match between all corresponding feature pairs.
7. Transform the features of the model according to the recovered transformation T and verify them against the input image features. If the verification fails, go back to step 2 and repeat the procedure using a different image basis pair.

Example – Recognition (1)

- Access hash table entries to find candidates for matching models

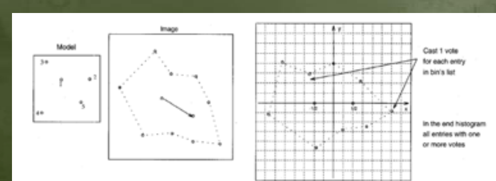
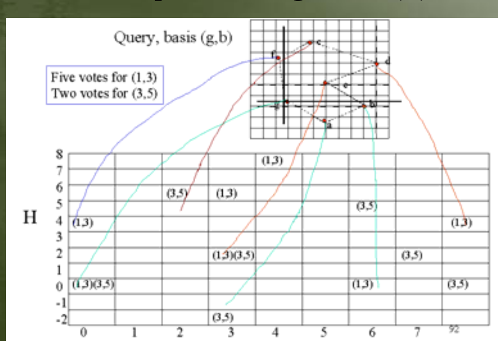


Figure 3. Determining the hash table bins that are to be notified when two arbitrary image points are selected as a basis. Similarity transformation is allowed.

Example – Recognition (2)



Complexity

Assume $m=n$, and k is the number of point to define the basis.

- Preprocessing: $O(n^{k+1})$ for a single model.
- Recognition: $O(n^{k+1})$ against all objects in the database.

Under Various Transformations (1)

- Translation in 2D and 3D.
  - 1-point basis.
  - O(n^2).
- Similarity transformation in 2D.
  - 2-point basis.
  - O(n^3).
- Similarity transformation in 3D.
  - 3-point basis.
  - O(n^4).

Under Various Transformations (2)

- Affine transformation
  - 3-point basis.
  - O(n^4)
- Projective transformation
  - 4-point basis.
  - O(n^5)

Recognition of 3D Objects from 2D Images (1/5)

1. Correspondence of planes
   - Preprocessing: consider planar sections of the 3D object which contain three of more interest points.
   - Hash (model, plane, basis) triplet.
   - Use either projective transformation or affine transformation.
   - Once the planes correspondence have been established, the position of the entire 3D body is solved.

Recognition of 3D Objects from 2D Images (2/5)

2. Singular affine transformation
   - $A\,x + b = U$ where
   - A : 2x3 affine matrix
   - x : 3x1 3D vector
   - b : 2x1 2D translation vector
   - U : 2x1 image

Recognition of 3D Objects from 2D Images (3/5)

3. A set of four non-coplanar points in 3D defines a 3D affine basis:
   - One point as origin
   - The vectors between origin and the other three points as the unit (oblique) coordinate system.

   - Preprocess the model points in this four-basis point.

Recognition of 3D Objects from 2D Images (4/5)

4. Recognition:
   - *Pick four points:* p_0, p_1, p_2, *and* p_3 *--> three vectors:* v_1, v_2, *and* v_3 *in the 2D image.*
   - *Exists:* $\alpha v_1 + \beta v_2 + \gamma v_3 = 0$, *where* $(\alpha, \beta, \gamma) \neq 0$
   - *A point* p *in the image, with* v *be the vector from* p_0 *to* p.
   - Vote for all $t \neq 0$ (*a line with parameter* t):

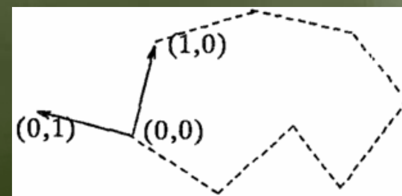
     $v = (\xi + t\alpha)\, v_1 + (\eta + t\beta)\, v_2 + (t\gamma)\, v_3$, where (ξ, η) is the coordinate of v in the v_1, v_2 basis.

Recognition of 3D Objects from 2D Images (5/5)

5. Establishing a viewing angle with similarity transformation.
   - Tesselate a viewing sphere (uniform in spherical coordinates).
   - Record (model, basis, angle) in the hash table.
   - 2-point basis: $O(n^3)$ (the same order as without viewing angle because the viewing angle introduces only a constant factor -- independent of the scene).

Recognition of Polyhedral Objects

- Polygonal objects
  - Choose an edge as the basis, record (model, basis edge) in the hash table.



  - Preprocessing and recognition is $O(n^2)$.

Index Distribution

- Probabilistic distribution model of points result in non-even bin sizes, and the length of the longest list dominates the time needed to carry out the histogram phase of the algorithm.

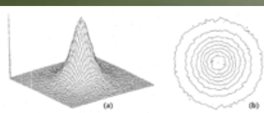


Figure 5. The distribution over the space of invariants, and several of its contours, for the case of Gaussian-distributed point features. Similarity transformation is allowed.

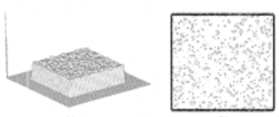


Figure 6. Hash table equalization for the case of similarity transformations and point features generated by the Gaussian process $N\left(0,\begin{pmatrix}\sigma & 0\\ 0 & \sigma\end{pmatrix}\right)$: (a) the expected distribution of remapped invariants; (b) several of the distribution's contours.

Noise Modeling

- Problem
  - Positional errors result in "wrong" has bin indices.
  - There is still good chance that the wrong bin is in the neighborhood of the correct bin.
- Possible Solutions
  - Rectangular bin access instead of a single bin
  - Weighted voting around neighborhoods

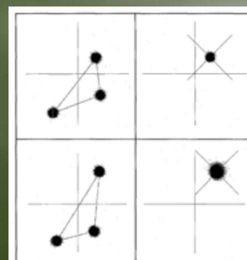


Figure 8. Regions of the hash table that need to be accessed when there is Gaussian error in the positions of the point features. Similarity transformation is allowed. The left graph of each pair shows the feature space domain, whereas the right shows the space of invariants. For presentation purposes, the amount of Gaussian error was deliberately made large.

Bayesian formulation

- Maximum likelihood approach
  - *Pr((Mk, i,j, B) | S')*
    - *Mk* is model *k*
    - *i,j* are points that match the reference set
    - *B* is the basis set
    - *S'* is the set of points calculated relative to *B*
  - Maximize:

$$\log(\Pr((M_k, i, j, B))) + \sum_{p_\xi \in S} \log\left(\frac{\Pr(p_\xi \mid (M_k, i, j, B))}{\Pr(p_\xi)}\right)$$

Conclusions

- The method is based on the representation of objects by point sets and matching corresponding sets of points
- By applying geometric constraints these sets of points can be further represented by a small subset of points (basis points)
- The size of the basis depends on the transformation applied to the models
  - A basis of 2 points is sufficient for 2-D scenes under rotation, translation and scale
  - A basis of 3 points is sufficient for affine transformation for the perspective view
- The process is divided into preprocessing and recognition – reduces complexity, enables off-line preprocessing

References

1. Papers
   1. Y. Lamdan and H. Wolfson, “Geometric hashing: A general and efficient model-based recognition scheme,” ICCV 1988
   2. H.J. Wolfson and I. Rigoutsos. “Geometric Hashing: An Overview.” IEEE Computational Science & Engineering, 1997.
2. Slides
   1. B.Prunomo, Geometric Hashing, Johns Hopkins Univ., 2004
   2. J.Costello, A.Kukuch, and T.Tatge, Geometric Hashing, Indiana University, 2006
   3. I.Eidhammer and I.Johassen, Protein Structure Comparison and Structure Patterns- an Algorithmic Approach, Tutorial, ISMB 2001
   4. Spring 2006 Visual Recognition, Lecture notes, Technion – Israel Institute of Technology