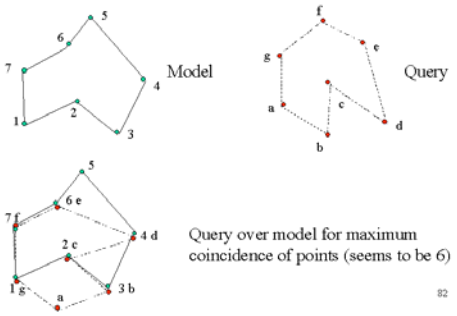
Geometric Hashing

- Originally, a Computer Vision technique for matching objects in a scene to models in a database.
- Based on the idea of storing invariant features of models in a hash table and finding matching features of the query object in this hash table.

Example

- Geometric hashing example for matching 2D figures (from Eidhammer's ISMB'01 Tutorial)
- Two figures are given, a **model** *A*, and a **query** *B*, described by *m* and *n* points, respectively
- Find common subfigures, invariant under rotation and translation (scale is not used)
- One approach is to "place the query on top of the model", and consider how many points coincide (*here we ignore the edges*)

Example



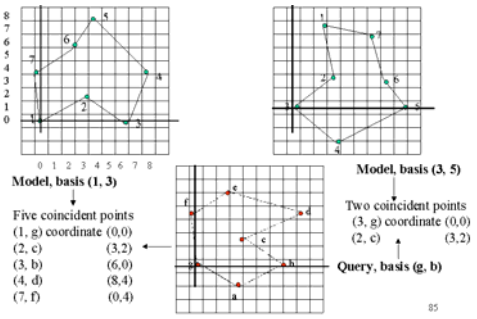
Coincidence sets

- Finding the maximal coincidence set is NP-hard
- If you are looking for biologically significant matching substructures of two proteins, you may want to find *all* coincidence sets with the number of pairs over a given threshold

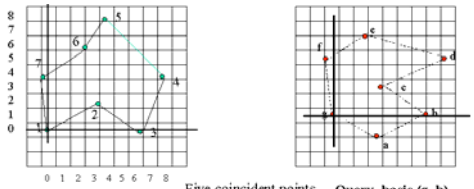
Reference frames

- 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 Systems



Example: Reference Frame Systems



Five coincident points	
(1, g) coordinate	(0,0)
(2, c)	(3,2)
(3, b)	(6,0)
(4, d)	(8,4)
(7, f)	(0,4)

Point 6 (2,6) not found coincident with point e (2,5)

Remarks

- 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.

Remarks

- 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.*

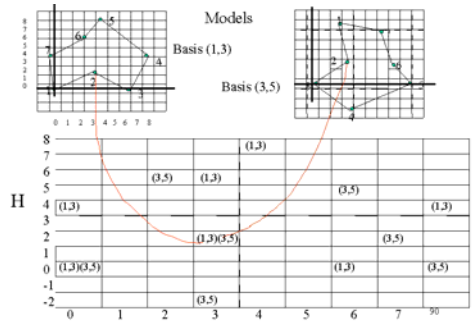
Hashing

- For dealing efficiently with all combinations, hashing is used. It is especially efficient when several queries are to be compared to one model, or to several models.
- The comparison problem can be formulated as: *given a query reference frame system, for each model reference system, in how many cells are there points from both the query and the model frame system?*
- The hashing technique makes it possible to simultaneously compare a query frame system to all model frame systems.

Preprocessing

- In this example, a 2D hash table *H* is used. It has a bin for each cell in the frame systems. In a **preprocessing** phase, the coordinates of all points in each **model** frame system are found. If there is a point in the cell (p,q) in the frame system with basis (a_i, a_k), then (a_i, a_k) is placed in the bin $H(p,q)$
- Since all pairs of points from the model will (generally) act as basis pairs, totally m*(m*(m-1)/2) pairs will be in *H*

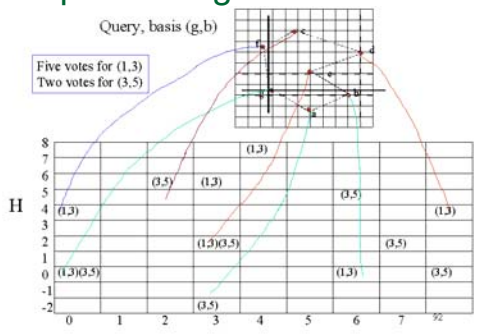
Example: Preprocessing



Recognition

- The **query** is compared to the model in the **recognition** phase
- A pair is chosen as basis, and the coordinates of the other points are calculated
- These coordinates are used as indices into H, and for each cell being indexed, a vote is given for the (model) basis pairs in the cell. The number of votes for a model basis pair is the number of coinciding points to the query (using the specified query basis pair)

Example: Recognition



Geometric Hashing for Protein Structure Comparison

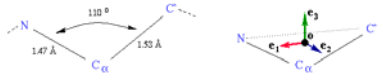
- How can we use geometric hashing to find matching substructures of two proteins?
- We can use 3 non co-linear amino acids (C_α coordinates) to define a reference coordinate frame.
- Then we can store the positions of the other amino acids in a 3D hash table.
- Do this for every possible triple of C_αs of the model in the preprocessing phase

Geometric Hashing for Protein Structure Comparison

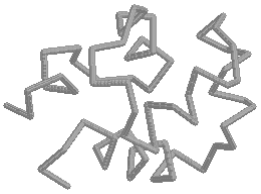
- In the recognition phase, we can do the same for each basis triple of the query protein and count the matching C_αs.
- Complexity analysis:
  - $O(m^4 n^4)$ for two proteins of size m and n.

Alternative technique by Pennec and Ayache

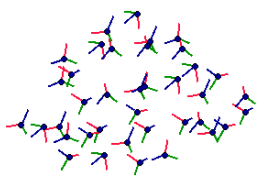
- Instead of using triples of amino acids we may also use a single amino acid to define a reference frame. This approach works because 3 atoms of an amino acid are always in the same geometric configuration.



Modeling the proteins



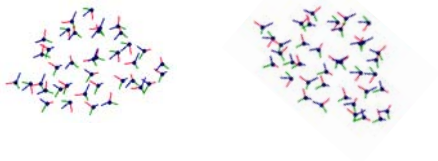
Backbone of the protein



Protein modeled as an unordered set of frames

Comparing protein structure

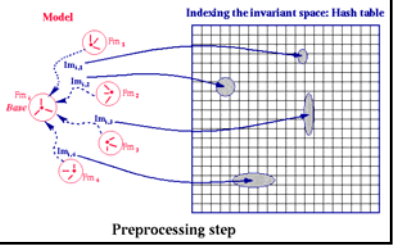
- To find similar substructures in two proteins, we now have to find two subsets of frames that are in the same configuration, up to a global rigid transformation.



Geometric hashing: Invariant Representation

- To obtain an invariant representation with respect to the global position and orientation of the protein, the configuration of all frames relative to the basis frame is used. This is stored in a 6D hash-table and for correctness, the uncertainty of each invariant is included.

This preprocessing step is repeated with each amino acid as the basis.



Geometric hashing: Recognition

- Each amino acid of the second is used as the basis in turn and matching frames are found. If the basis belongs to a common substructure, then a significant number of frames are in the same configuration with respect to it.