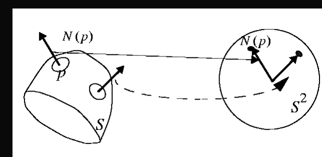
Spherical Attribute Images (SAI)

Goal

- Define a representation of a 3D model that is canonical
- Independent of translation, rotation

Previous Works

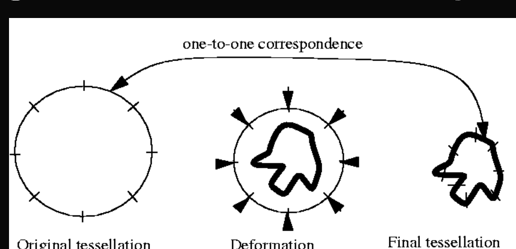


[Ikeuchi, Hebert 95]

- Gauss Map
- Extended Gaussian Image (EGI)
  - Associate mass with mapping
  - Can’t handle non-convex object
- Variants of EGI

SAI

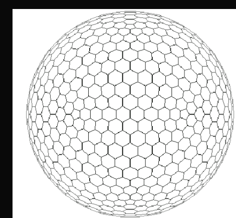
- Deform a semi-regularly tessellated geodesic dome onto the object



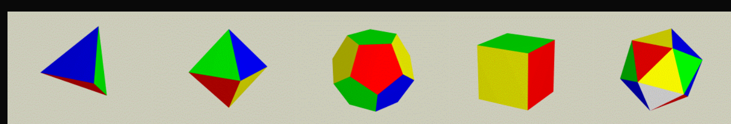
[Ikeuchi, Hebert 95]

Semi-regular Tessellation

- Subdivide icosahedron into $20N^2$ triangular faces
- Take the dual
- Every vertex have 3 neighbors



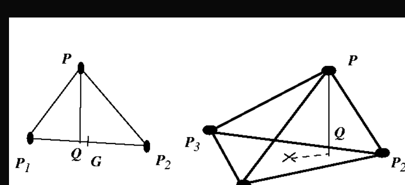
[Ikeuchi, Hebert 95]



Platonic Solids, [Gettys]

Surface Mapping

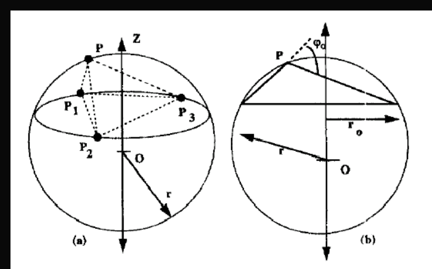
- Data Force
- Regularity Constraint
- Iterative deformation



Regularity [Ikeuchi, Hebert 95]

Curvature

- Approximated by simplex angle
- Ranged from $-\pi$ to $+\pi$



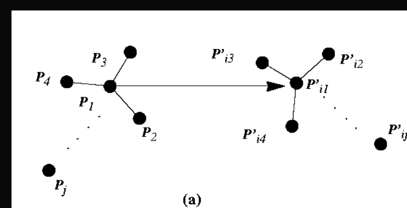
[Ikeuchi, Hebert,Delingette 95]

Properties

- For given number of nodes, invariant to translation and scaling
- Unique SAI for an object up to a rotation
- Connected patch of surface map to connected patch of the spherical image

Matching of two SAI

- Define $D(S,S',R) = \sum (g(P_s) - g(RP_{s'}))^2$
- Naïve : Try all possible rotations (θ,φ,ψ)
- Smart : Only 3K valid rotations for K nodes



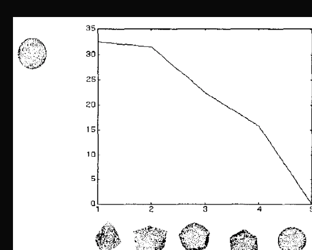
[Ikeuchi, Hebert 95]

On similarity

$$d_p(S,S',R) = \left(\int \left| g(P_S) - g(RP_{S'}) \right|\right)^{\frac{1}{p}}$$

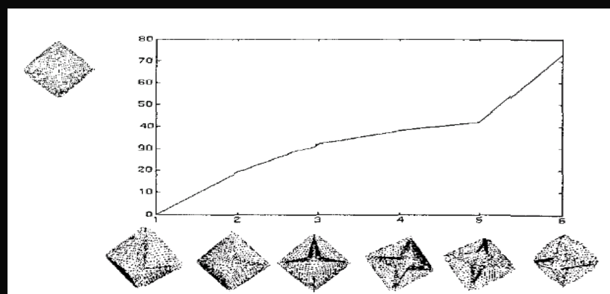
$$D_p(S,S') = \min_R d_p(S,S',R)$$

- D is metric
  - D(A,B) ≥ 0
  - D(A,A) = 0
  - D(A,B) = D(B,A)
  - D(A,B)+D(B,C) ≥ D(A,C)



[Shum, Ikeuchi, Hebert 96]

Similarity



[Shum, Ikeuchi, Hebert 96]

Discussion

- Good for matching identical objects
- Applied to merging multiple views, recognition of known surfaces
- Similarity…
- Restricted to genus zero

Point Set Matching

Surface Matching

- Extract "features" (usu. points, maybe labeled)
- Do point matching
- Lots of 2D algorithms, few extend to 3D

Allowed Transformations

- Translation + rotation = **rigid motion** (Euclidean)
- + scaling = similarity
- + shearing = affine

Exact Point Matching

- Numerically very unstable
- 2D:
  - get polar coords w.r.t. centroid
  - sort by φ, r & concatenate => “strings” A,B
  - is A substring of BB ?
  - O(n log n) <- sort
- For similarity, first scale object diameter

Exact Point Matching : 3D

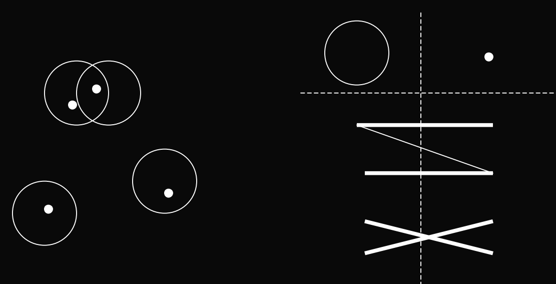
- Project points onto center sphere
- Label projections w/ list of projected pts.
- Take 3D convex hull of points on sphere to induce adjacency relationship
- Solve labeled planar graph isomorphism
- O(n log n)

Approximate Matching

- Bound Hausdorff distance
- One – to – one mapping
- Can compute optimum or exact answer, but it’s a real pain

Approximate Matching

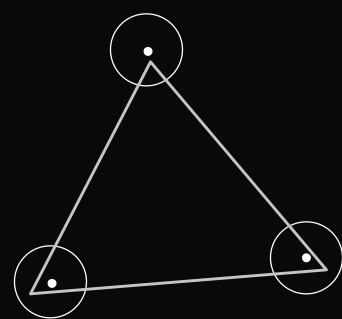
- If point sets are “close”, is it 1-to-1?
- Network flow on bipartite graph



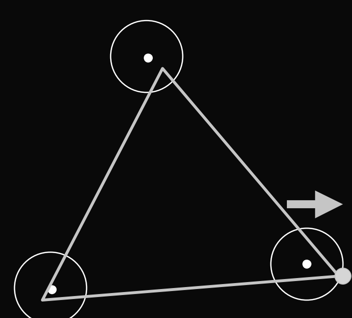
Approximate Matching

- Find rigid transformation
- Idea : in 2D, 2 points must satisfy Hausdorff distance exactly (3 points in 3D)

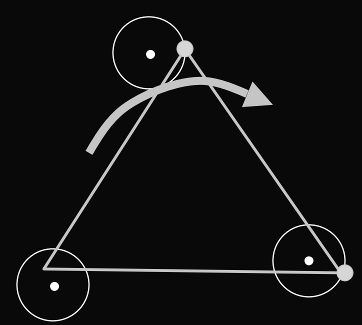
Approximate Matching: Idea



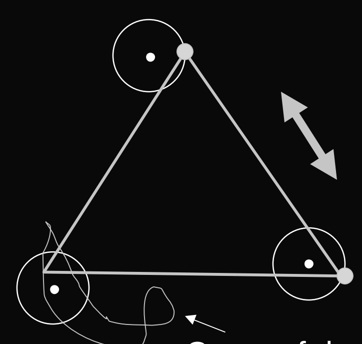
Approximate Matching: Idea



Approximate Matching: Idea



Approximate Matching: Idea



Approximate Matching

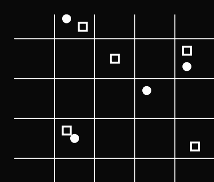
- Curve has at most 12 intersections with circle (n-2 curves, n-2 circles => $O(n^2)$)
- Try all possible 2-2 initial pairing: $O(n^8)$
- 3D : fix 3 points … ugh!

Not One-to-one

- "Pattern matching" (find "object" into "scene")
- Solution within constant factor of optimum (rigid motion in 3D: $8+\varepsilon$)
  - Pick 3 "opposing" points in object
  - Match to every triplet of scene points
  - Keep configuration w/ min Hausdorff distance

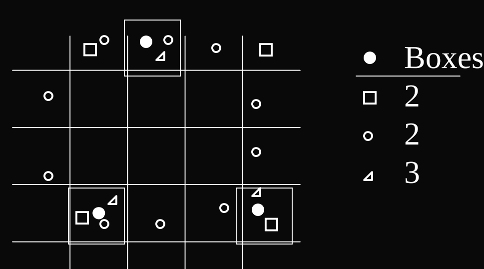
Acceleration Techniques

- For each *, find closest □. O(nm)
- Use grid & look only in appropriate box. O(n)



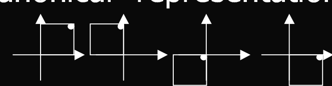
Geometric Hashing

- Goal 1: database query (no transformation)
- In all boxes with *, vote for shapes present



Geometric Hashing

- Goal: identify object & recover transformation
  - Objects in database appear multiple times, in all "canonical" representations



  - Ex: for translation, 1 "interest point" => origin

Geometric Hashing

- Canonical "basis"
  - 2D affine basis = 3 "interest points"
  - 3D rigid motion: 3 points
- Properties
  + Deals with occlusion
  + Highly parallel
  - Large table (memory requirements) $O(mn^4)$