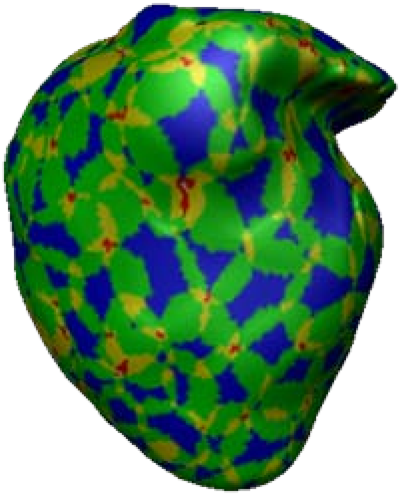
Constructive manifolds for surface modeling

Cindy Grimm

Washington University in St. Louis
Media and Machines
Computer Science Department

Representing shape

- What kinds of shapes?
  - Organic, free-form
  - Medical imaging
- How to specify? User interface?
  - Sketching
  - Adaptive
  - Hierarchical
- Why manifold approach?
  - Build in pieces
  - Design machinery to blend pieces



Overview

- What does it mean to be manifold? What is an atlas?
  - Disks, charts, overlaps, transition functions
- Traditional atlas definition
  - Building an atlas for a spherical manifold
- Surface modeling
  - Hierarchical and adaptive surface construction
  - Reconstruction from scattered data points
    - Consistent parameterization
- Constructive definition
  - Building abstract manifolds
    - Parameterizing implicit surfaces

Manifold (adj.)

- A surface is manifold if it is locally Euclidean
  - Pick a point. Grow neighborhood around point (disk)
  - Can deform disk to R^n (no tearing, folding)
- Atlas: cover manifold with disks
  - Disks overlap

Can deform to separate self-intersections

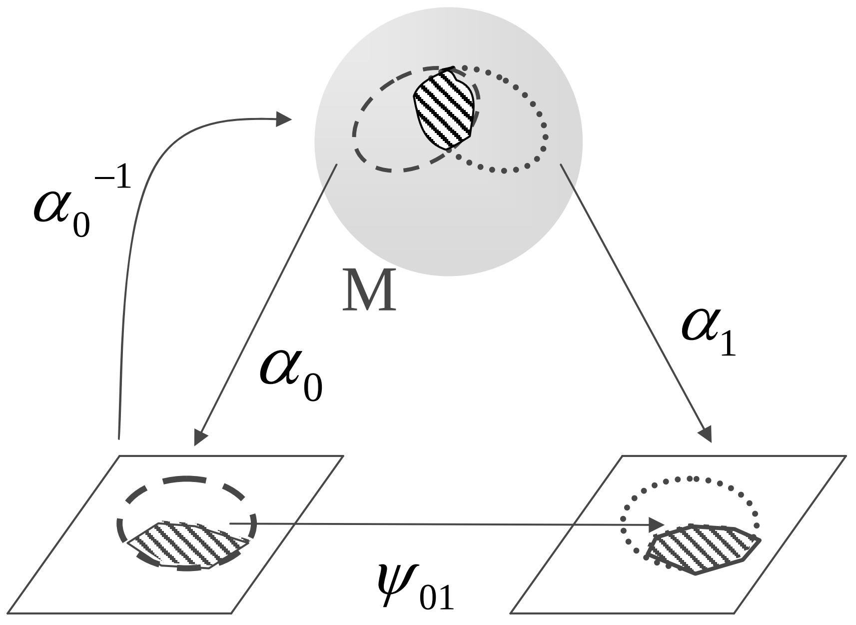
Traditional atlas definition

Given: Manifold *M*

Construct: Atlas *A*

- Chart
  - Region U_c in M (open disk)
  - Region c in R^n (open disk)
  - Function α_c taking U_c to c
    - Inverse
- Atlas is collection of charts
  - Every point in M in at least one chart
  - Overlap regions
  - Transition functions:

  $\psi_{01} = \alpha_1 \circ \alpha_0^{-1}$ smooth



Manifold (constructive)

- Build a surface in pieces
  - Pieces overlap
  - Continuity by transition functions
- How to build overlaps/transition/chart functions?
  - How do you ensure continuity? Correctness? Topology?
  - Can't just overlap at random
  - Computationally tractable
- What are good local maps (charts)?

- Add geometry

Atlases for spherical manifolds

- Uses:
  - Surface modeling
  - BRDF, environment maps
- Challenges:
  - No non-singular global parameterization
  - Tools exist for operating on the plane
    - E.g., spline functions
- Implementation:
  - How to represent points?
  - How to specify charts?
    - Overlaps? Transition functions?

Points on the sphere

How to represent points on a sphere?

- Problem:
  - Want operations (e.g., linear combinations) to return points on the sphere

All points such that…

$$x^2 + y^2 + z^2 = 1$$

- Solution: Gnomonic projection
  - Project back onto sphere
    - Valid in ½ hemisphere
  - Line segments (arcs)
  - Barycentric coordinates in spherical triangles
    - Interpolate in triangle, project

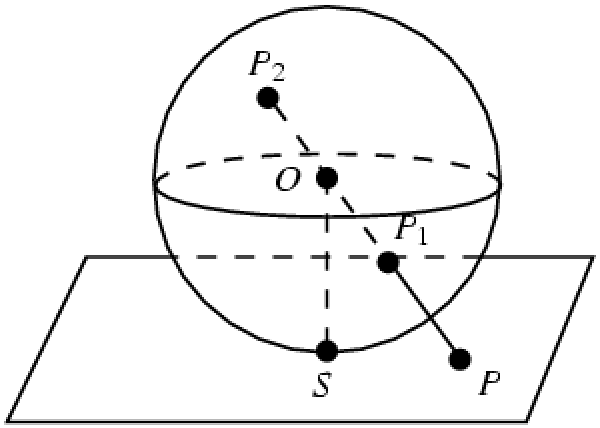
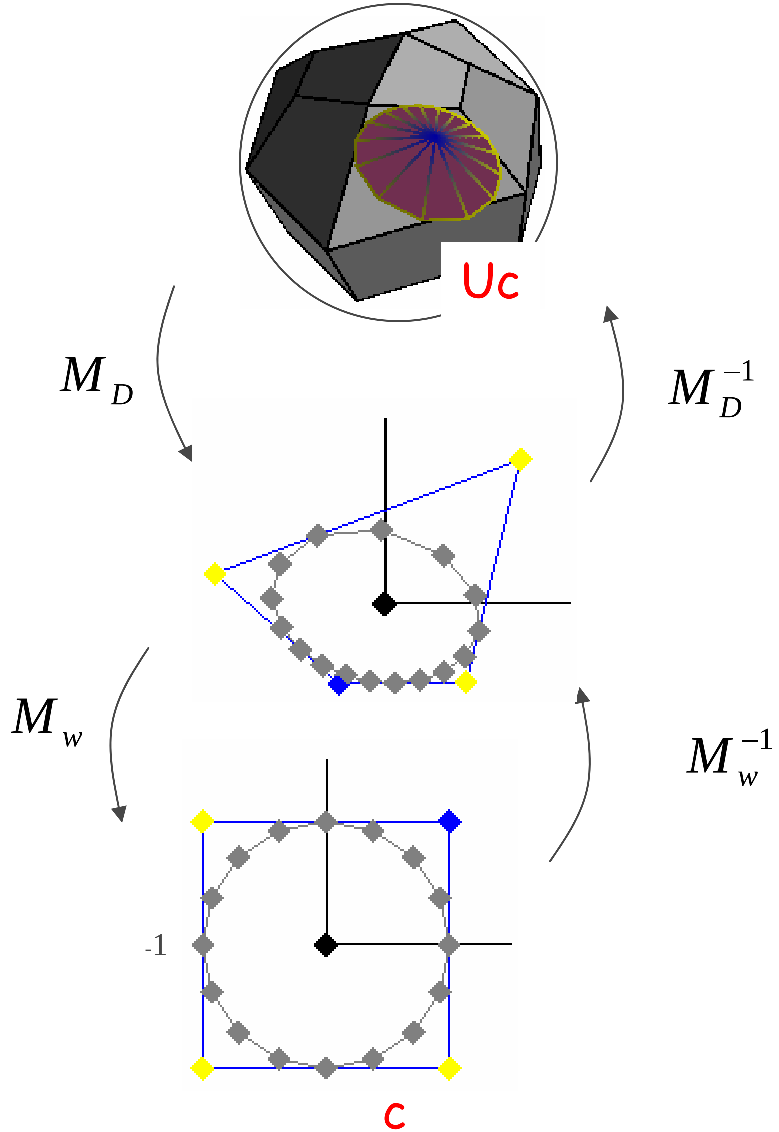


Chart on a sphere

Chart specification:

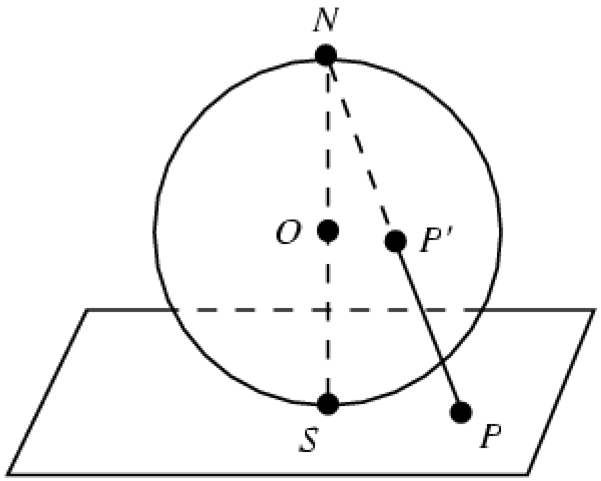
- Center and radius on sphere U_c
- Range c = unit disk
- Simplest form for α_c
  - Project from sphere to plane
    - (optional) Adjust

$$\alpha_c = M_W(M_D(x, y, z))$$



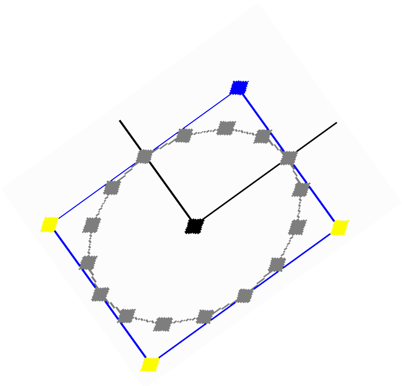
Projection to plane

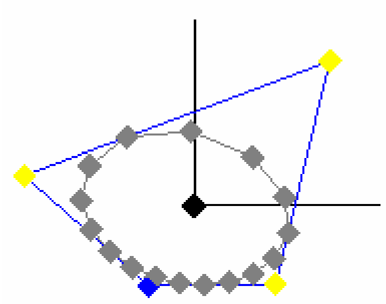
- Multiple choices (busy map makers)
  - Invertible
  - Preserve local geometry
  - Not biased by projection point
  - Analytic
- Stereographic
- Orthographic
- Latitude Longitude
  - Rotate sphere before projection

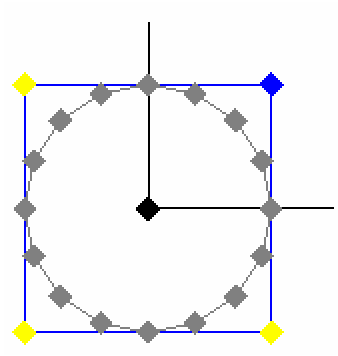


Warp after projection

- Better control over area of projection
  - Invertible
  - Analytic
  - Affine map
    - Projective map







Washington University in St. Louis
Media and Machines
Computer Science Department

Defining an atlas

- Define transition functions, overlaps, computationally
- Point in chart: evaluate α_c
- Coverage on sphere (U_c domain of chart)
  - Define in reverse as $\alpha_c^{-1}=M_D^{-1}(M_W^{-1}(D))$
    - D becomes ellipse after warp, ellipsoidal on sphere
    - Can bound with cone normal
  - Transition function is

$$\psi_{ij} = \alpha_j \circ \alpha_i^{-1}$$

Why do I care?

- Charts
  - Avoids global parameterization problem
    - Get chart over area of interest
    - Run existing code as-is
      - **Optimize position**
- Atlas
  - Embed sphere using existing techniques
    - E.g., splines, polynomials
      - **No special boundary cases, e.g., duplicated end points, geometric constraints**

Writing functions on manifolds

Do it in pieces

- Write embed function per chart
    - Can use any R^n technique
        - Splines, Radial Basis Functions, polynomials…
        - Doesn’t have to be homogenous!
- Write blend function per chart
    - k derivatives must go to zero by boundary of chart
        - Guaranteeing continuity
    - Normalize to get partition of unity
        - *Spline functions get this for free*

Final embedding function

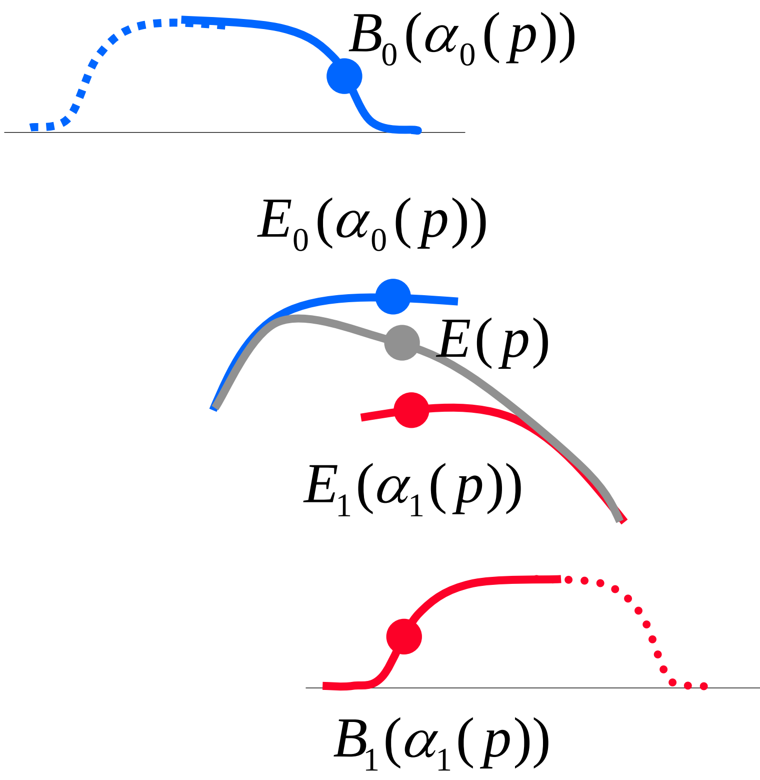
- Embedding is weighted sum of chart embeddings
  - Generalization of splines
  - Given point p on manifold
    - Map p into each chart
      - **Blend function is zero if chart does not cover p**

Map each chart

Embed

$$E(p) = \sum_{c \in A} B_c(\alpha_c(p)) E_c(\alpha_c(p))$$

Blend



Continuity is minimum continuity of constituent parts

Alternative formulation

- Define using transition functions
  - Derivatives
  - Pick point p in a chart
    - Map p to all overlapping charts

Embed

$$E(\alpha_c^{-1}(p)) = \sum_{c' \in A} B_{c'}(\psi_{c'c}(p)) E_{c'}(\psi_{c'c}(p))$$

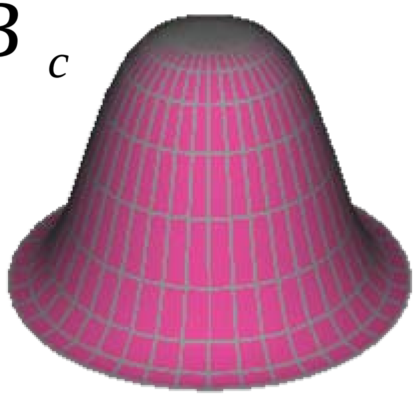
Blend

Partition of unity

- Blend function for each chart
  - B-spline basis spun around origin
  - Divide by sum to normalize

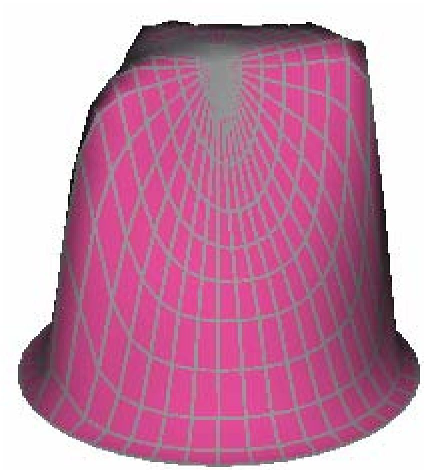
$$B_c(p) = \frac{\hat{B}_c(\alpha_c(p))}{\sum_{c' \in A} \hat{B}_{c'}(\alpha_{c'}(p))}$$

$\hat{B}_c$



Proto blend function

B_c



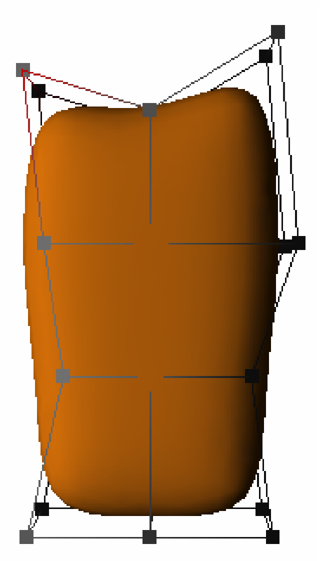
Normalized blend function

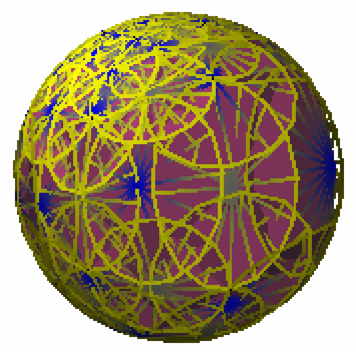
Embedding

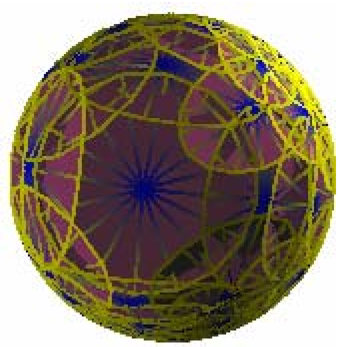
- Each chart has own embedding function
  - Nice if they agree where they overlap
- Polynomials in these examples
  - Simple
  - No end conditions, knot vectors…

Surface editing

- User sketches shape
  - Subdivision surface
- Create charts
  - One chart for each vertex, edge, and face
- Fit each chart to subdivision surface (locally)

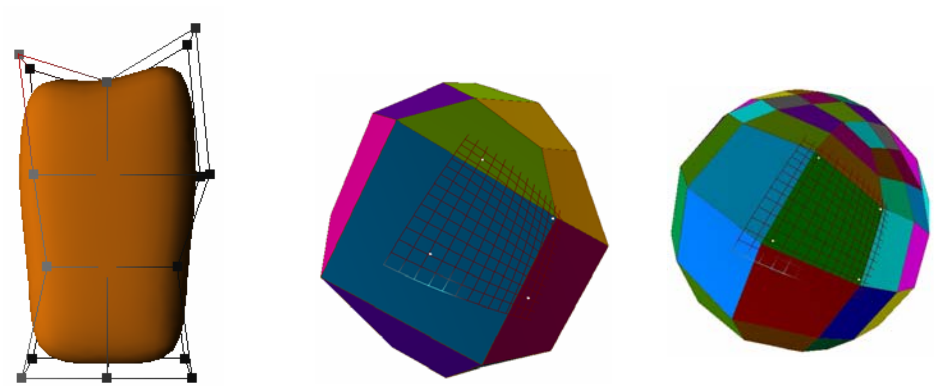






Construction approach

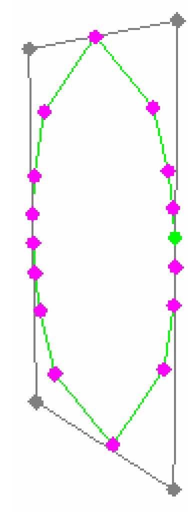
- Simultaneously embed subdivision surface in sphere
  - Any algorithm works; regularity
- Always maintain 1-1 relationship between surface, subdivision mesh, and sphere
  - Note: no geometric smoothing on sphere
    - Vertex->vertex, edge->mid-point, face->centroid



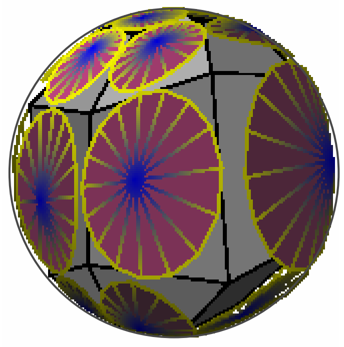
First level subdivision

Charts

- Optimization
  - Cover corresponding element on sphere
  - Don't extend over non-neighboring elements
- Projection center: center of element
  - Map neighboring elements via projection
  - Solve for affine map
- Face: big as possible, inside polygon
  - Use square domain, projective transform for 4-sided



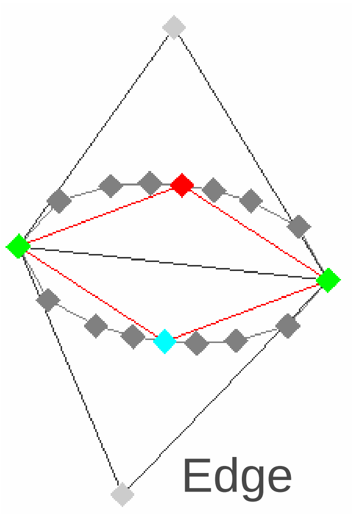
Face

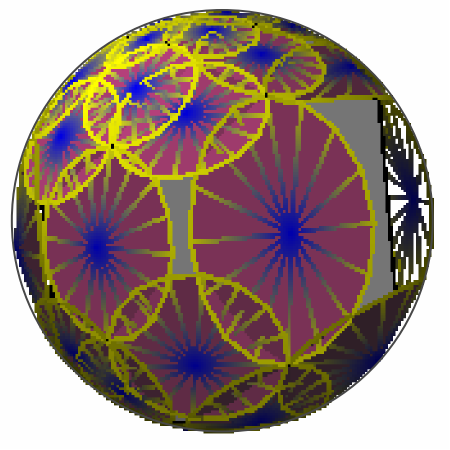


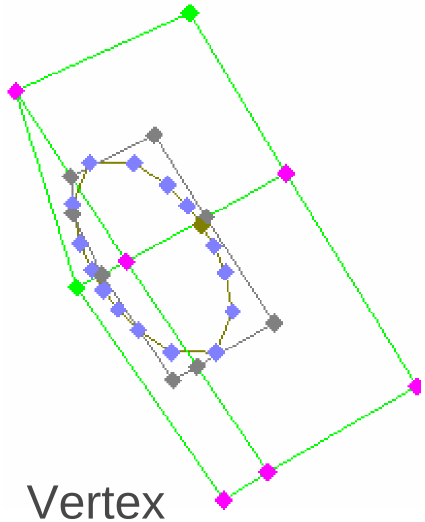
Face charts

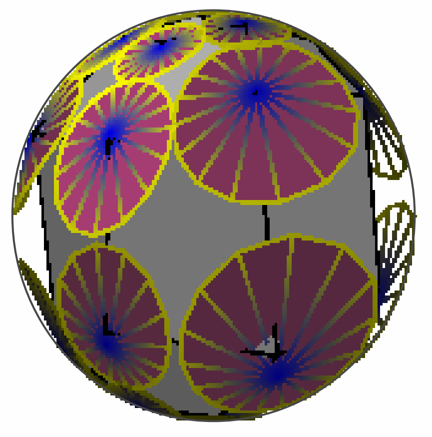
Edge and vertex

- Edge: cover edge, extend to mid-point of adjacent faces
- Vertex: Cover adjacent edge mid points, face centers


Edge


Edge charts


Vertex


Vertex charts

Coverage

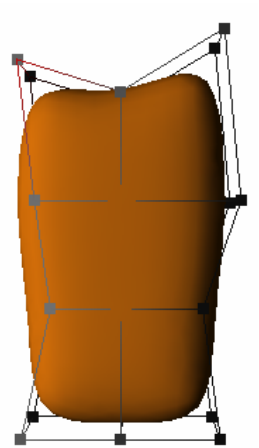
- Can adjust to optimize overlap
  - Guarantee minimal overlap
- Optimize 2 or 3 overlaps
- Open question: What is a good chart arrangement?

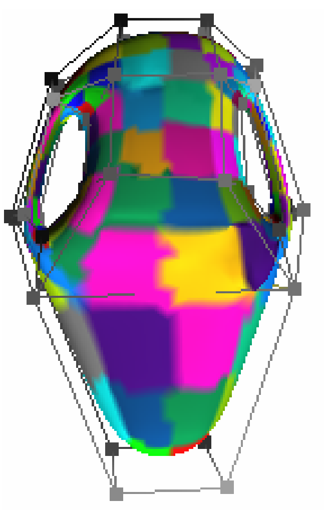
Making surface embedding

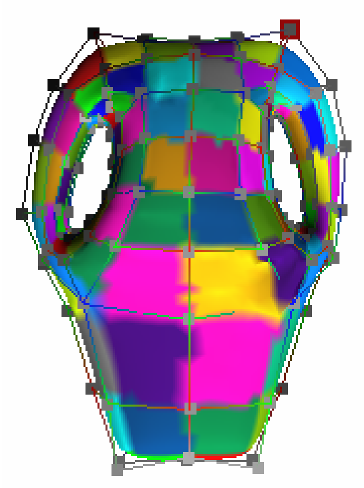
- Fit each chart embedding to subdivision surface
  - Least-squares $Ax = b$
- 1-1 correspondence between surface and sphere
  - Generate grid of points in chart
    - Chart to sphere to point in subdivided mesh (3 times) to determine (u,v) in face
  - Generate subdivision surface point
    - (Jos Stam, Exact Evaluation)

Summary

- C^K analytic surface approximating subdivision surface
- Real time editing
  - Works for other closed topologies
    - *Parameterization using manifolds, Cindy Grimm, International Journal of Shape modeling 2004*

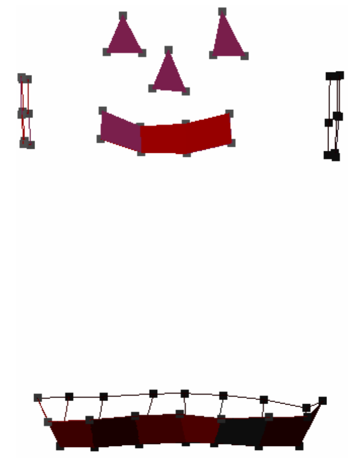


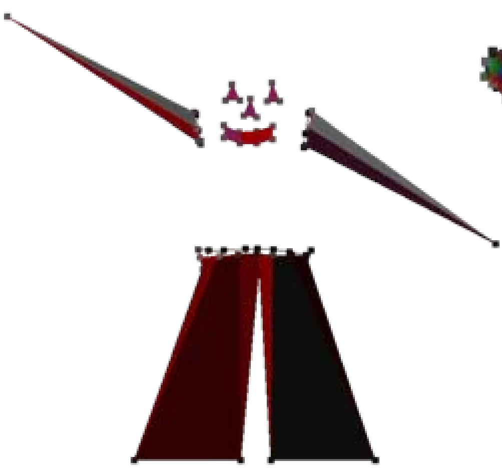


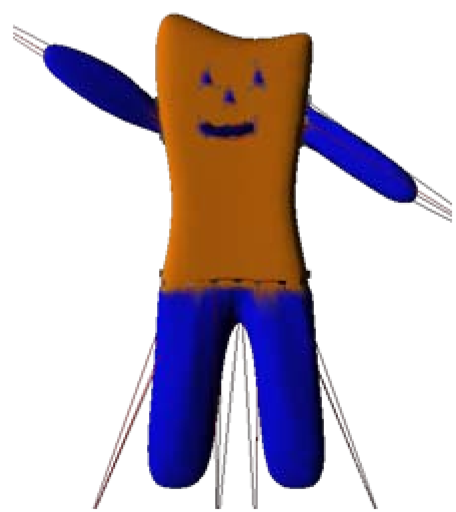


Hierarchical editing

- Override surface in an area
- Add arms, legs
    - User draws on surface
    - Smooth blend
    - No geometry constraints

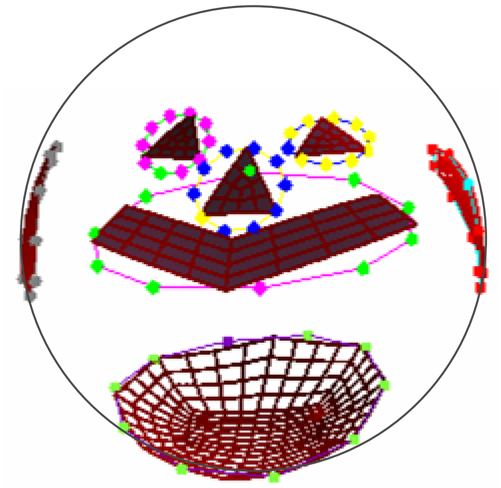






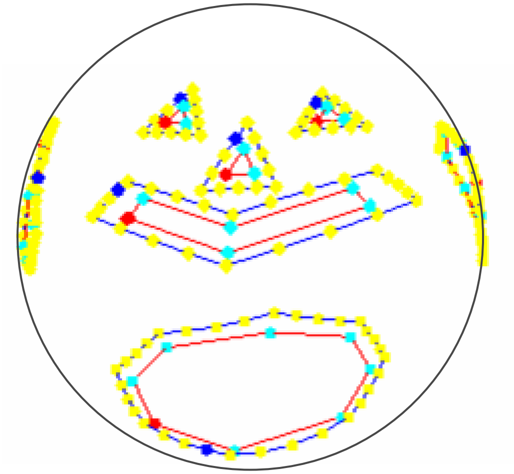
Adding more charts

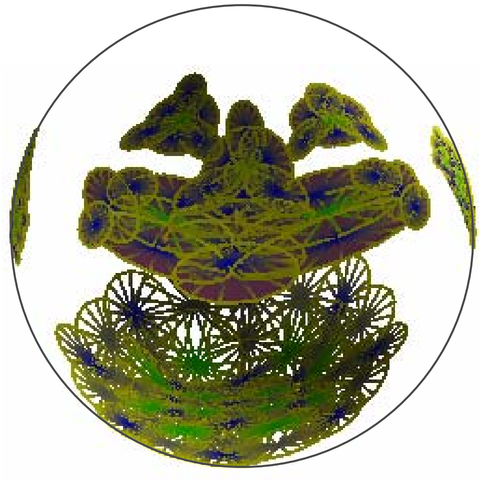
- User draws new subdivision mesh on surface
    - Only in edit area
- Simultaneously specifies region on sphere
    - Add charts as before
- Problem: need to mask out old surface



Masking function

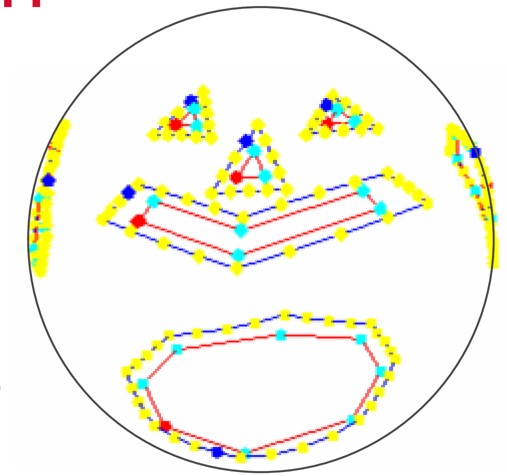
- Alter blend functions of current surface
  - Set to zero inside of patch region
- Alter blend functions of new chart functions
  - Zero outside of blend area
- Define mask function η on sphere,
  - Set to one in blend region, zero outside

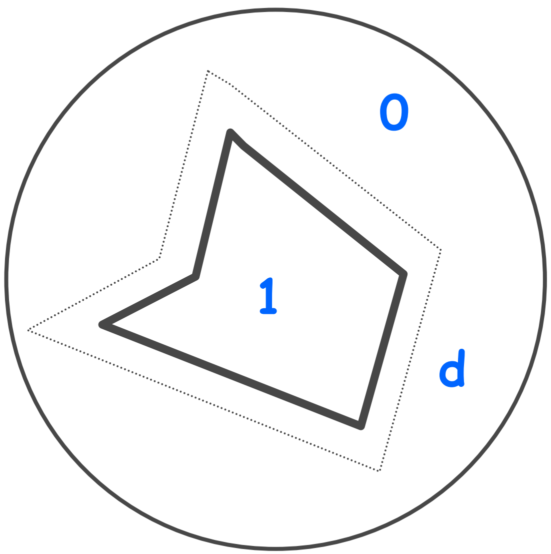




Defining mask function

- Map region of interest to plane
    - Same as chart mapping
- Define polygon P from user sketch in chart
- Define falloff function f(d) -> [0,1]
    - d is min distance to polygon
    - Implicit surface
- Note: Can do disjoint regions





Washington University in St. Louis
Media and Machines
Computer Science Department

Patches all the way down

- Can define mask functions at multiple levels
- Charts at level *i* are masked by all *j>i* mask functions

$$\hat{B}_c(p) = \Pi_{j>i}(1-\eta_j(p))B_c(\alpha_c(p))$$

- Charts at level *i* zeroed outside of mask region

$$\hat{B}_c(p) = \Pi_i \eta_i(p) B_c(\alpha_c(p))$$

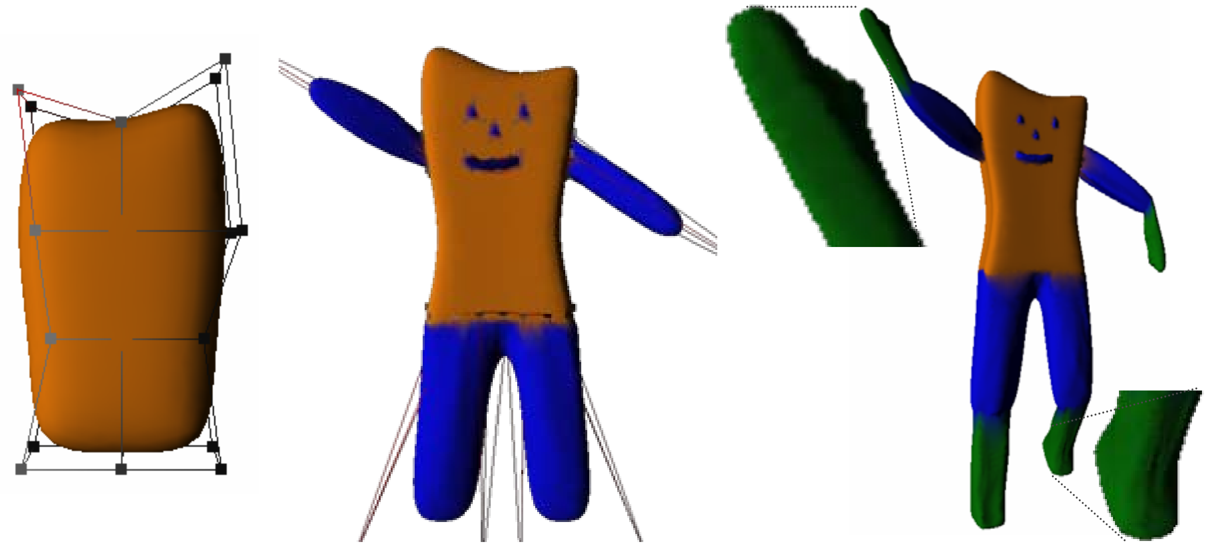
Creases and boundaries

- Introduce chart embedding function with discontinuity
- Mask out overlapping charts
- Incomplete mask results in smooth crease
- Boundary: curve in chart

Summary

- Flexible modeling paradigm
  - No knot lines, geometry constraints
- Not limited to subdivision surfaces
  - Alternative editing techniques?

Cindy Grimm, Spherical manifolds for adaptive resolution surface modeling, Graphite 2005

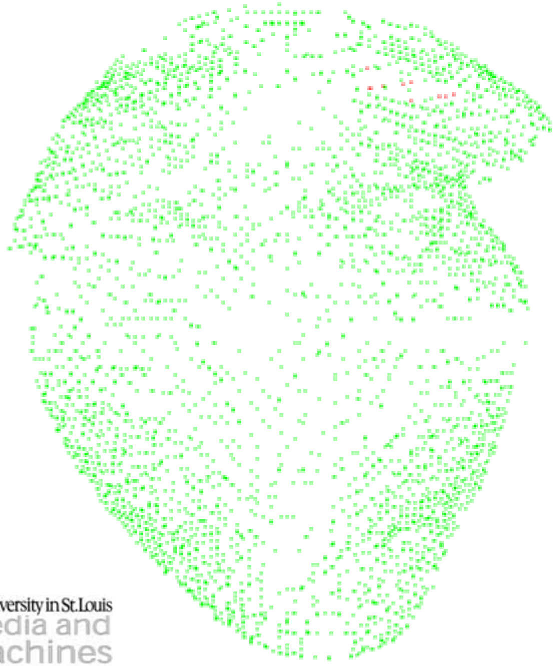


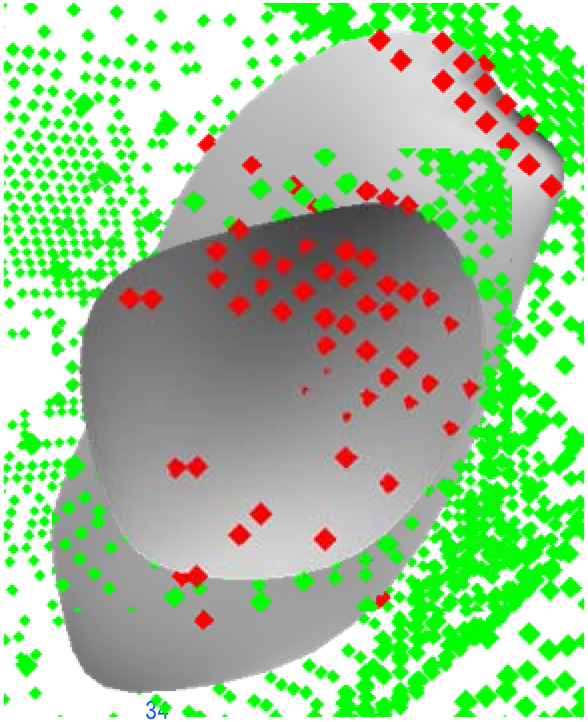
Open questions

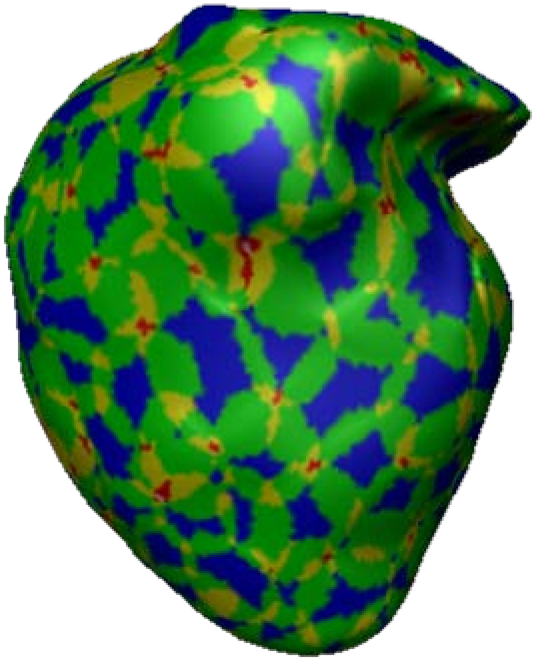
- What's the best chart placement strategy?
    - Number of charts?
- Tessellation
    - Tessellate domain, edge swap, move to centroids
- Better mask function
    - Concave, curved shape
- Moving between topologies
    - Topological surgery
- Better editing

Surface reconstruction

- Input: Scattered data, topology of surface
  - Non-uniformly sampled
  - Gaps
- Output: Smooth, analytical surface with correct topology



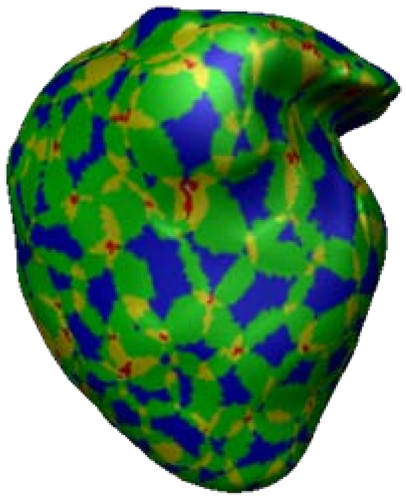


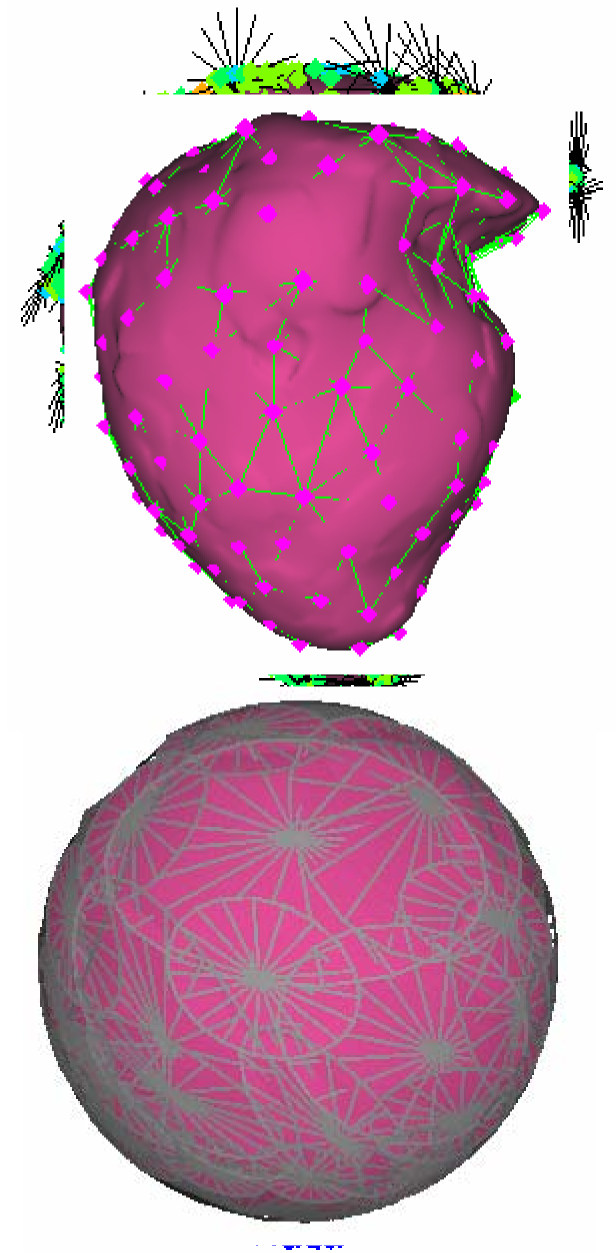


Washington University in St. Louis
Media and Machines
Computer Science Department

Outline

- Cover data with chart groupings
  - Multiple chart groups per point
- Embed chart structure on sphere
- Embed data points on sphere
- Make charts on sphere
- Fit charts to data



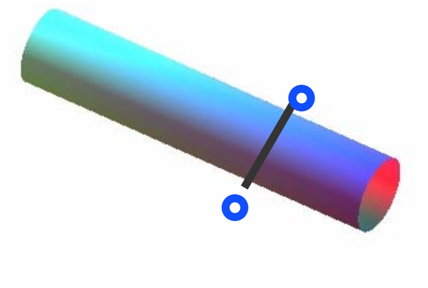


Considerations

- Avoid closest point, projection, iteration
- “Perfect” local neighborhood reconstruction
    - Avoid feature finding
- Approximation and extrapolation
    - Gaps in data

Local data topology

- Build tangent plane neighborhood and normal for each point
  - Try not to “cross” surface
  - Get neighbors in all directions
  - Ok to get wrong occasionally
- Algorithm
  - Search large neighborhood (min 10)
    - Stop when min angle < ¾ π
    - Ignore points in same direction, further out



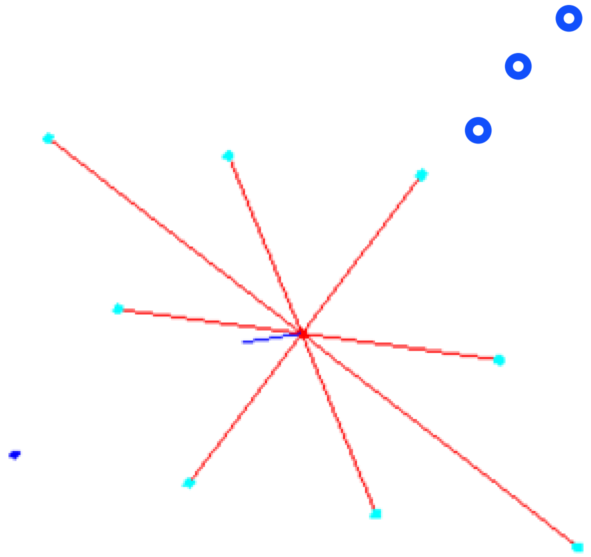


Chart groupings

- Based on geodesic distance on data point graph
- Seed point, radius r
  - All points within distance r
  - Dijkstra's shortest path
- Place new seed points at 2r-g from existing seed points
  - g = 0.3 is overlap
  - Hexagonal pattern

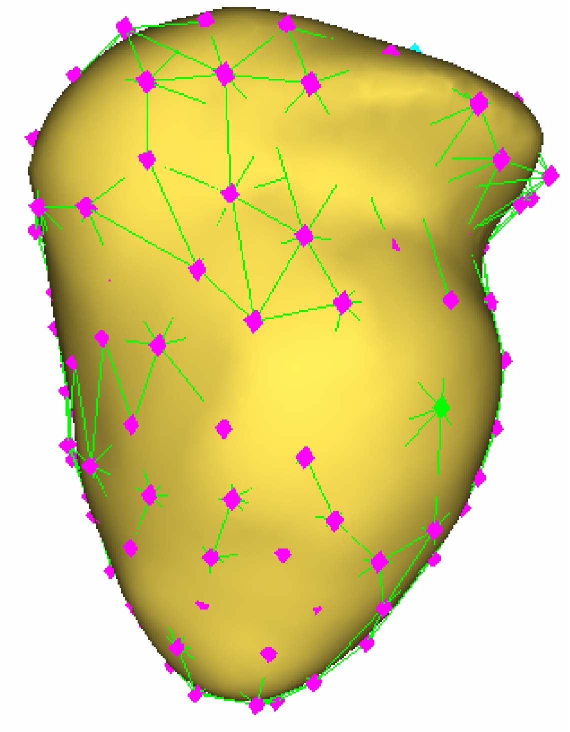
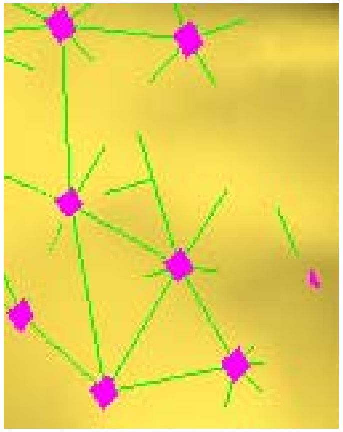
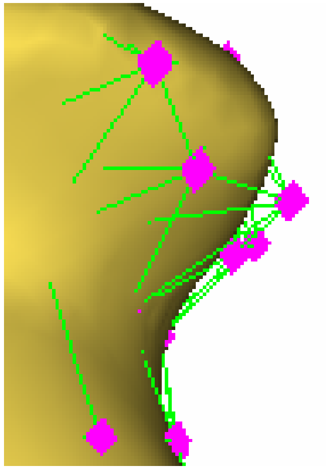


Chart connectivity

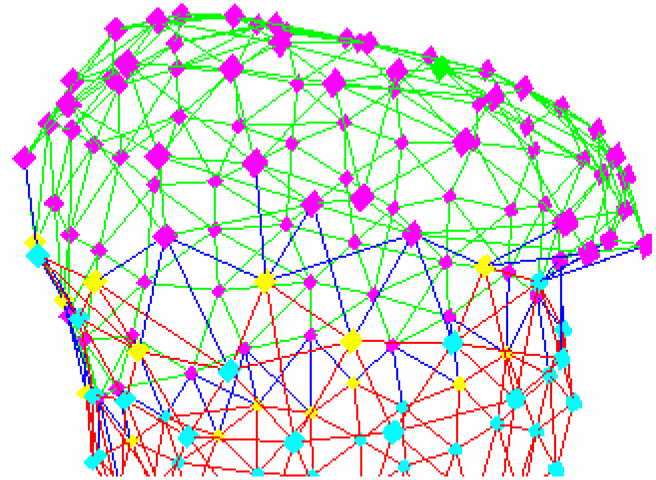
- Tangent plane neighborhood
  - Geodesic distance
  - Angle of geodesic in plane

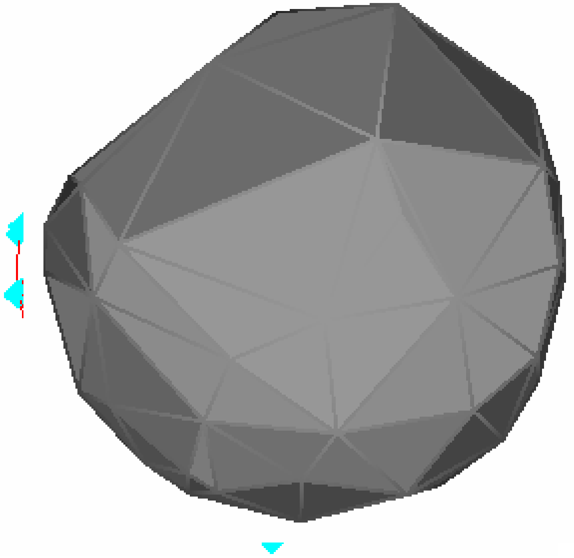




Embedding on sphere

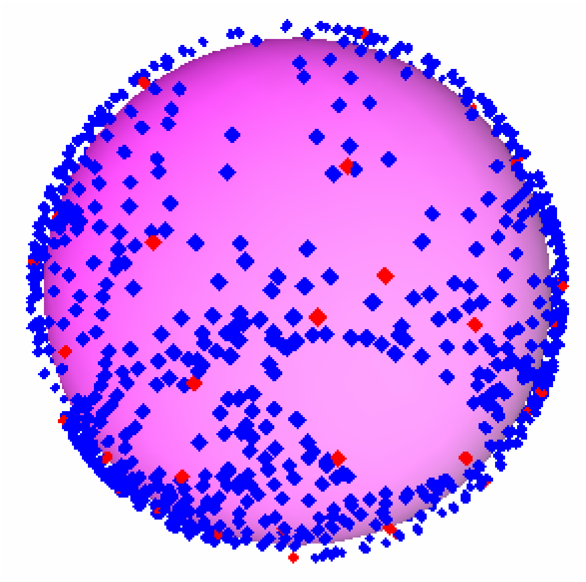
- Embed chart seed points on sphere
- Only have connectivity (no mesh)
    - Split in half, embed each half on a hemisphere
        - Floater parameterization
    - Adjust (convex hull to get mesh)
        - Move toward center





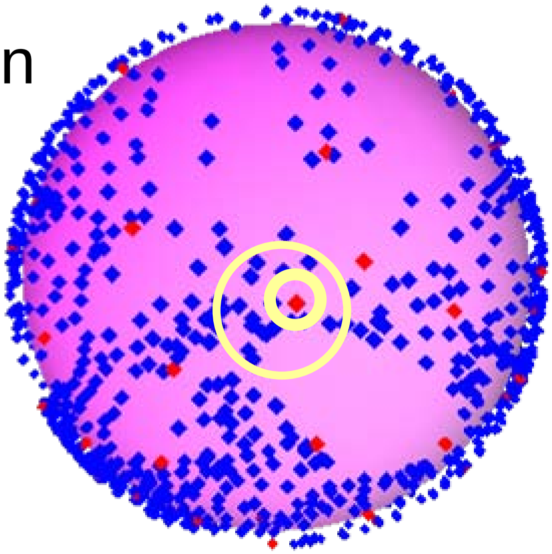
Embedding data points

- Fix seed points $S_j = (x,y,z)$
- Each data point goes to the center of its neighbors N_i
  - $\text{Sum}_i\ w_i\ N_i = p$
  - Least-squares $Ax = b$
- Project back onto sphere
  - *No meshing*



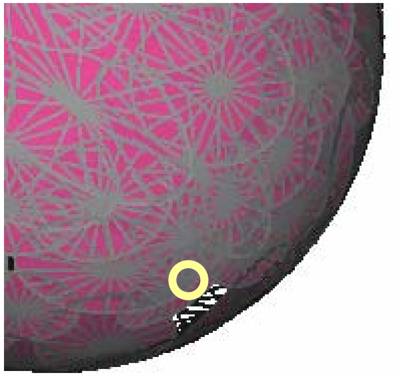
Making charts

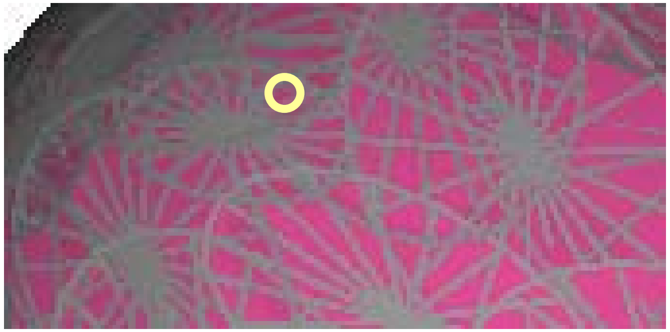
- Seed point location defines center of stereographic projection
- Partition data points
  - Assign to closest seed point (geodesic)
  - Add one ring of neighbors
- Chart must cover corresponding partition



Ensuring overlap

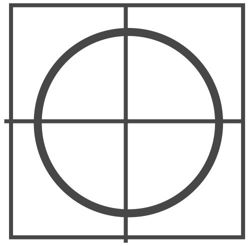
- Check atlas along boundary of chart for places that aren't covered
  - Add to coverage set
- Points on boundary that are well-covered
  - Remove from coverage set

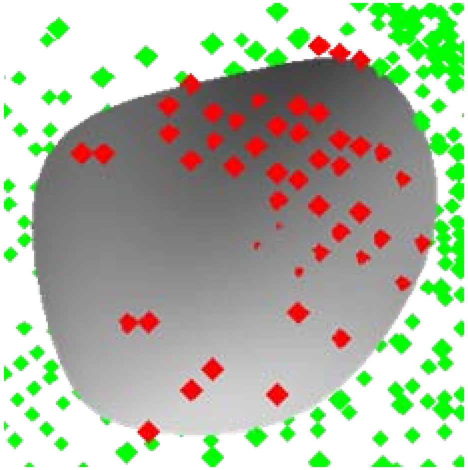




Fitting

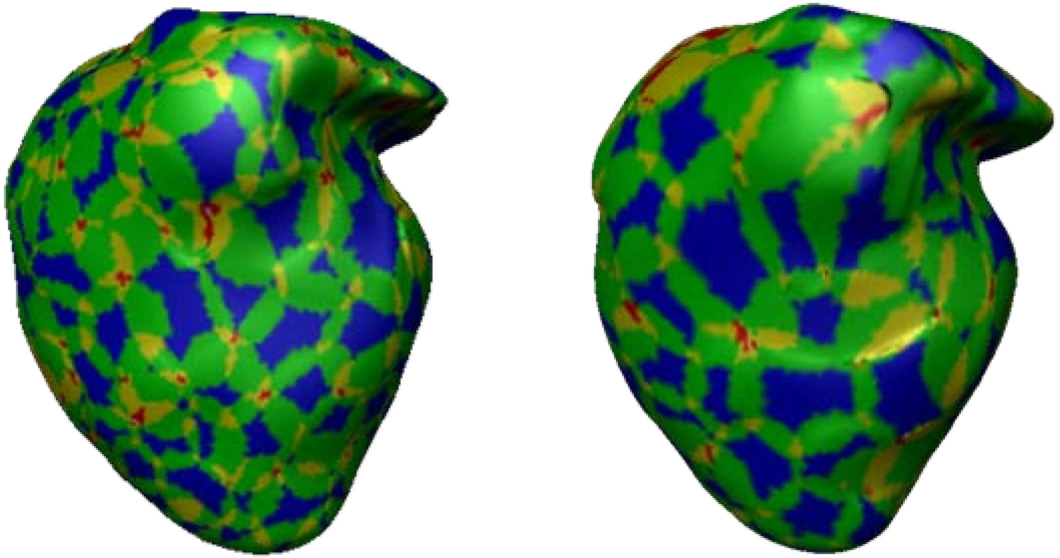
- Fit each chart individually
- All points in chart's domain, plus neighbors
- Smoothing terms
  - Second derivative is zero
  - First derivatives are same
- Least squares
  - Increase order until good fit (user-supplied average fit, percentage covered)





Fitting, smoothing

- Data points in chart
- Fit to existing surface along boundary of chart
- Reduce smoothing term
  - Makes chart embeddings agree along overlap areas



Results

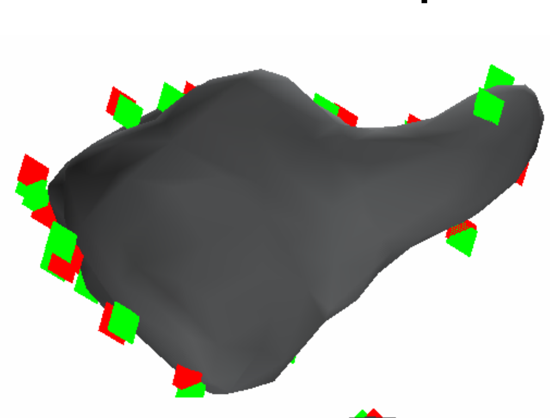
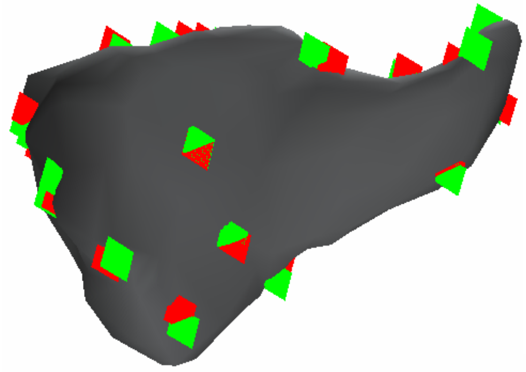
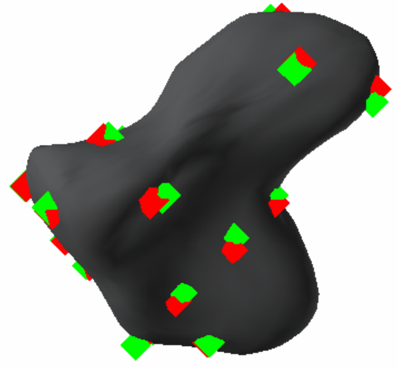
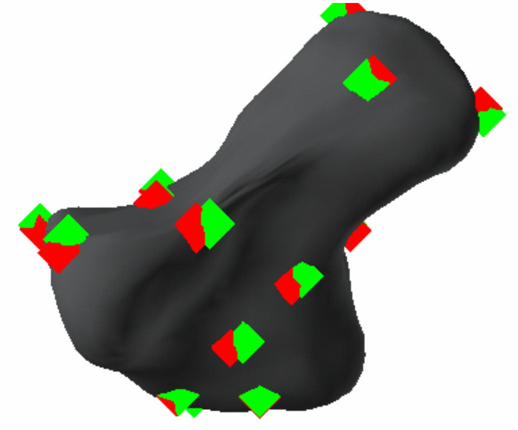
- Guaranteed topology
    - Data may “fold” onto sphere
- Should work with other topologies
    - Embed onto correct domain (no mesh)
- Doesn’t rely on correct neighborhood
    - Delay meshing as late as possible
    - No closest point

Results

- Open questions
  - Chart alignment, size, placement
  - Scattered data fitting
    - Error guarantees
    - Edge conditions
  - Detecting folding
    - Guarantees/conditions on original data?

Consistent parameterizations

- Problem: given same bone from left and right hand, match parameterization
  - Fix seed points (constrained parameterization)

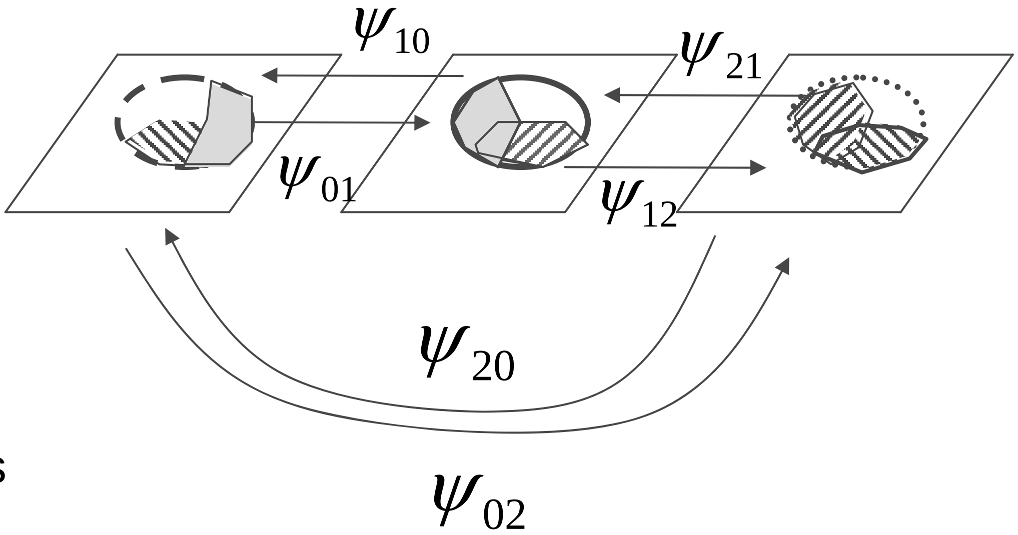





Joint with David Laidlaw, Joseph Crisco, Liz Marai, Brown University

What if we don't have a sphere?

- Previous approach relied on having an existing manifold
  - Cover manifold with charts
- Suppose you want to make a manifold from scratch?
  - Create manifold object from disks and how they overlap
  - Think of someone handing you an atlas; you can "glue" the pages together where they overlap to re-create the manifold
  - Resulting object is an *abstract* manifold
    - Requires some care to ensure glued-together object is actually manifold

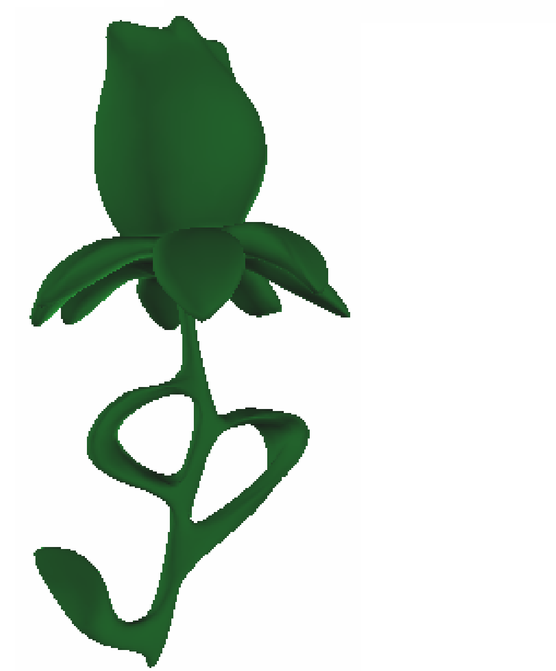
Constructive definition



Construct: Proto-Atlas *A*

- Disks
  - Region c in R^n (open disk)
- Overlap regions between disks
  - $U_{cc'} \subset c$
- Transition functions between disks
  - $\psi_{cc'}$
- To create a manifold object from A:
  - "Glue" points together that are the same, i.e.,
  - $\psi_{cc'}(p)=q$ implies p == q
- Transition functions must make sense
  - Reflexive: $\psi_{cc}(p)=p$
  - Symmetric: $\psi_{c'c}(\psi_{cc'}(p))=p$
  - Transitive: $\psi_{ik}(p)= \psi_{ij}(\psi_{jk}(p))$

From meshes



Cindy Grimm and John Hughes,
"*Modeling Surfaces of Arbitrary Topology using Manifolds*", Siggraph '95

J. Cotrina Navau and N. Pla Garcia,
"*Modeling surfaces from meshes of arbitrary topology",,* Computer Aided Geometric Design, 2000

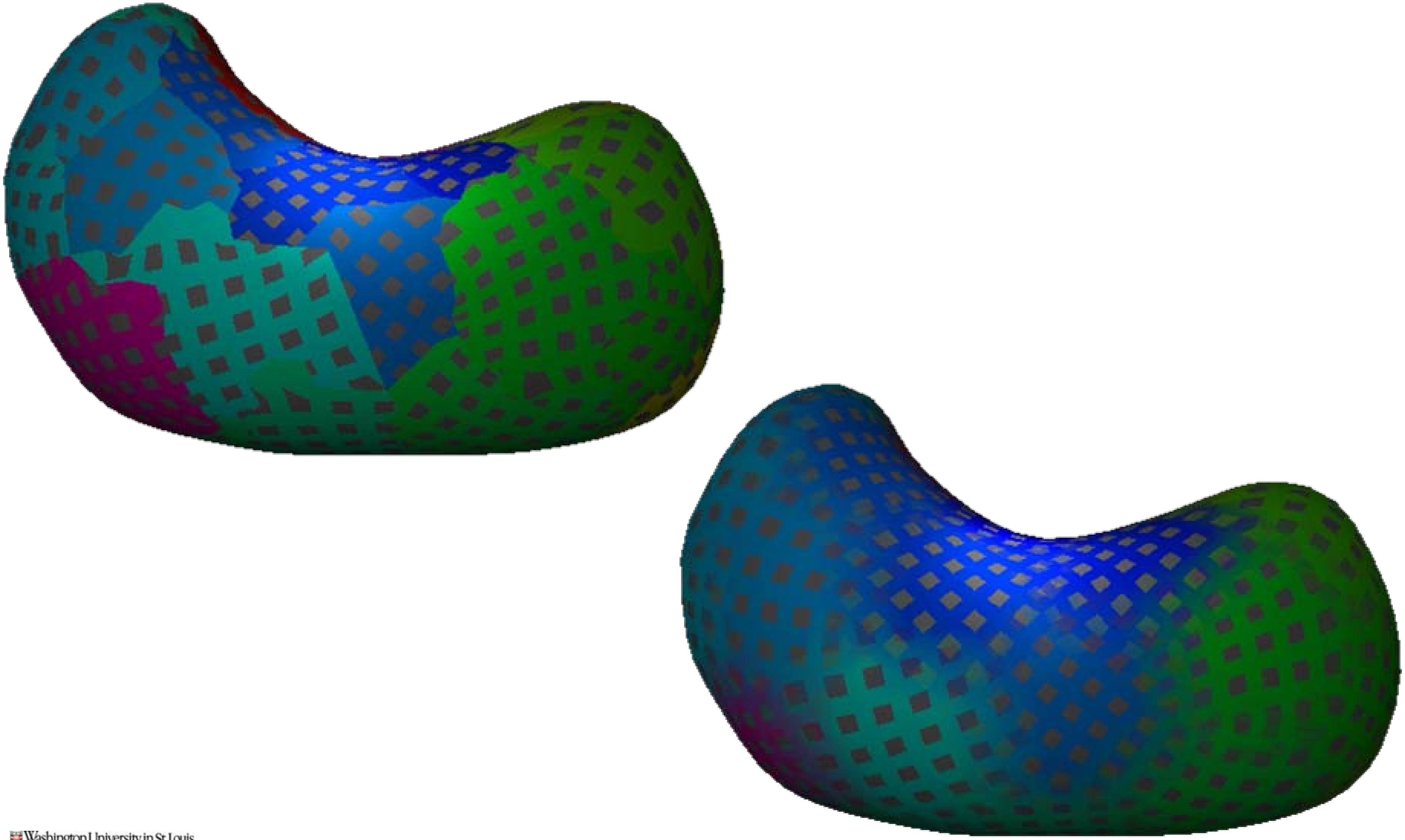
Lexing Ying and Denis Zorin,
"*A simple manifold-based construction of surfaces of arbitrary smoothness*", Siggraph '04

Xianfeng Gu, Ying He, and Hong Qin
Manifold Splines, ACM Symposium on solid and physical modeling

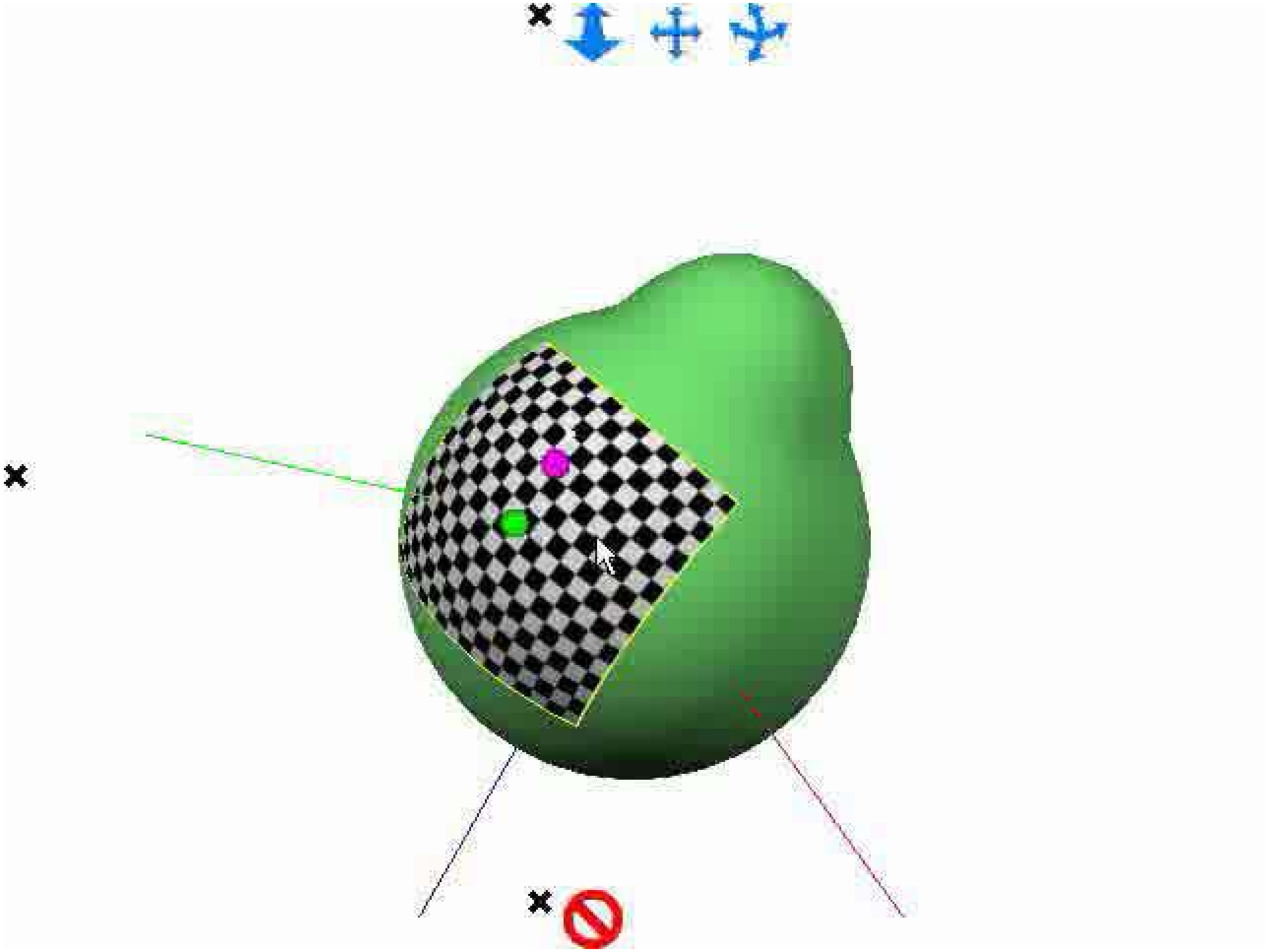
Parameterizing implicit surfaces

- Goals
  - Texture mapping for blobby objects
  - Robust to re-meshing, movement
- Build: Abstract, affine parameterization
  - Tessellation
- Basic idea
  - Repeat chart groupings from before
  - Parameterize each chart grouping
    - UV coords: "Bary coords" in 3D or projection
- Joint with Brian Wyvill & Ryan Schmidt (Univ. of Calgary)

Parameterization



Parameterization



Summary

- There are some manifolds we use often
  - Sphere, tori, circle, plane, S^3 (quaternions)
- Construct a general-purpose manifold + atlas + chart creation + transition functions
  - Now can use any tools that operate in R^n
    - *Use same tools for all topologies*
  - Can build charts at any scale, anywhere
    - Not dictated by initial construction/sketch

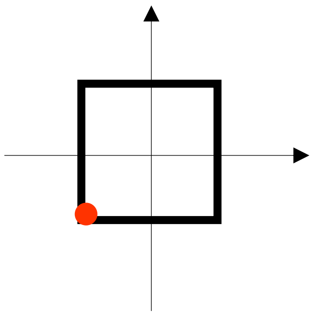
Open questions

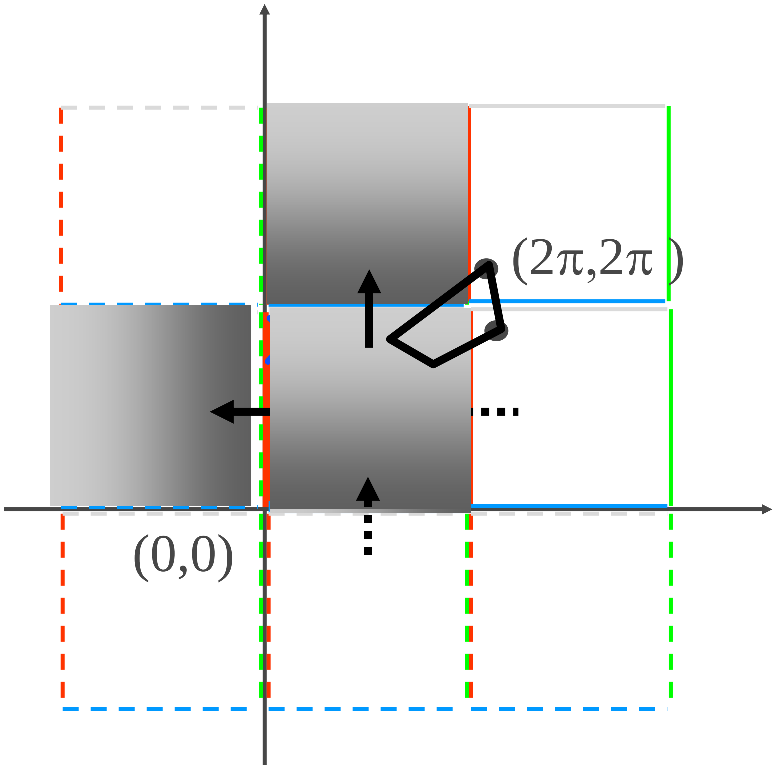
- Non-surface manifolds
- Boundaries
- Function discontinuities
- Changing topologies
- Establishing correspondences between existing surfaces and canonical manifold
    - Parameterization
- Where and how to place charts

Questions?

Torus

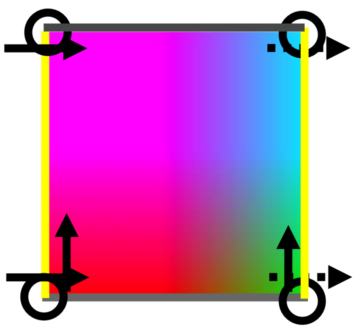
- Similar to circle example
  - Repeat in both s and t: [0,2π) X [0,2π)
  - Chart is defined by projective transform
    - Care with wrapping

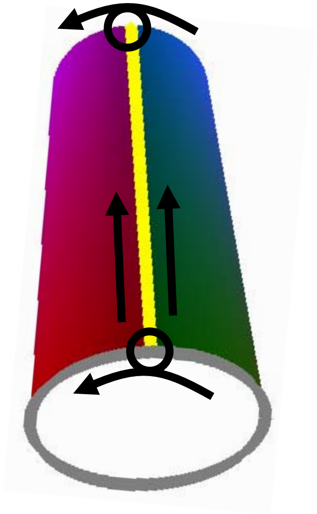


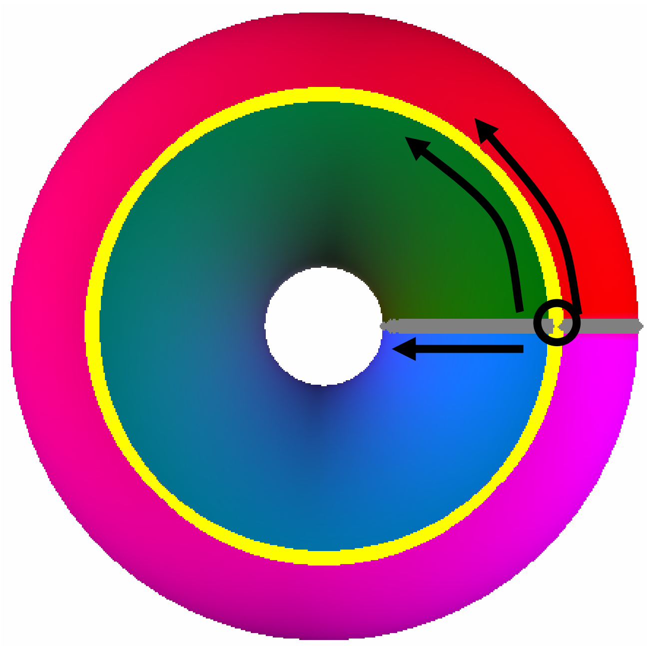


Torus, associated edges

- Cut torus open to make a square
  - Two loops (yellow one around, grey one through)
    - Each loop is 2 edges on square
    - Glue edges together
  - Loops meet at a point

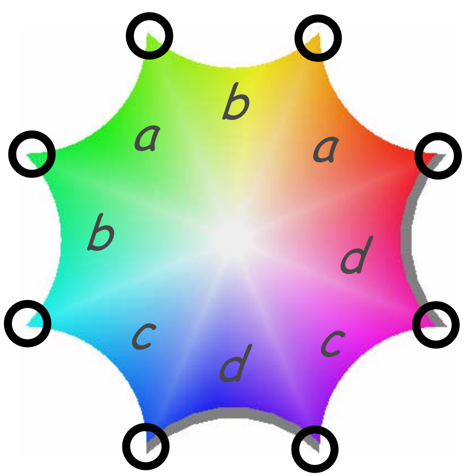


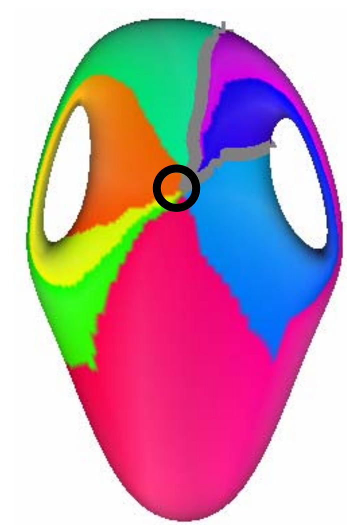




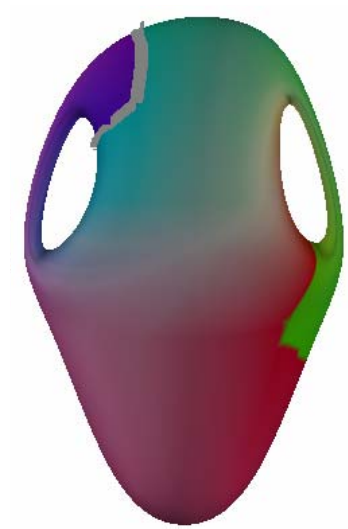
N-Holed tori

- Similar to torus – cut open to make a 4N-sided polygon
  - Two loops per hole (one around, one through)
    - Glue two polygon edges to make loop
  - Loops meet at a point
    - Polygon vertices glue to same point





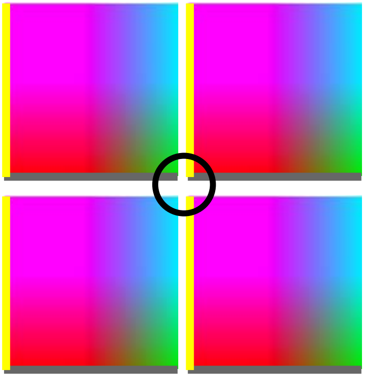
Front

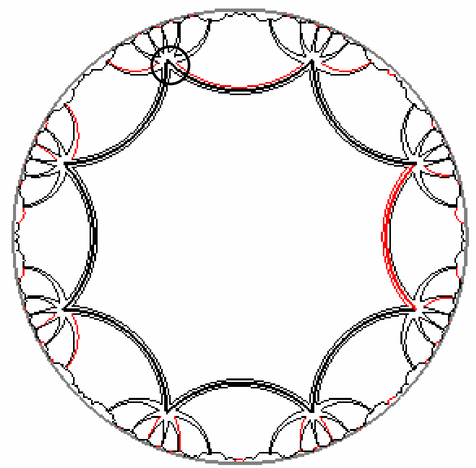


Back

Hyperbolic geometry

- Why is my polygon that funny shape?
  - Need corners of polygon to each have 2π / 4N degrees (so they fill circle when glued together)
  - Tile hyperbolic disk with 4N-sided polygons





Washington University in St. Louis
Media and Machines
Computer Science Department

Hyperbolic geometry

- Edges are circle arcs; circles meet boundary at right angles
- Linear fractional transforms
  - Equivalent to matrix operations in Euclidean geometry, e.g., rotate, translate, scale
  - Invertible
- Chart: Use a Linear fractional transform to map point(s) to origin, then apply warp function
  - Need to ensure we use correct copy in chart function