Surface Parameterization

History, etc
Introduction

• Hi, I am Ryan – http://www.dgp.toronto.edu/~rms / @rms80
• Head of Design & Fabrication Research at Autodesk until last October
• Developed Autodesk Meshmixer
  • Lots of mesh processing tools, based on lots of GeomProc research
• Now starting a “geometry processing startup” – gradientspace.com

• Open-source mesh-processing library geometry3Sharp (C#)
  • https://github.com/gradientspace/geometry3Sharp
  • Remeshing!
Parameterization

- Assigning a $M$-dimensional coordinate system to an $N$ dimensional space, where $M \leq N$

- Examples:
  - 3D space curve to real line
  - Hexahedral Element to Unit Box
  - Arbitrary genus-$N$ 3D shape to $N$-dimensional Torus
Surface Parameterization

• $N = 3, M = 2$

• NURBS Patch is canonical example

• **Parameter Domain** is a 2D rectangle

• **Embedding** is a smooth 3D surface
Parameter Domains are Useful
Parameter Domains are Useful
A small note on terminology

• “Parameter” is a pretty generic word

• When someone who is not a computer graphics person says “Parameterization”, even in a 3D context, they almost never mean Surface Parameterization like we will discuss today

• For example, if you hear a CAD person talking about “Parameterizing a Mesh”, or “Re-Parameterization”, they are talking about NURBS stuff
Mesh Parameterization
Texture Mapping

• Have square in image space
• Have 2D coordinates \((u,v)\) for each 3D triangle \((“uv” \text{ coordinates})\)

• For a point \(P\) in triangle:
  • Compute barycentric coordinates of \(P\) in 3D triangle
  • Reconstruct \(P\) in 2D triangle
  • Sample image
What else can you do with a parameterization?

• Remeshing
• Mesh Editing
• Geometric Computation
• Shape Analysis
• ...
• ...
• ...
• (Solve nearly any geometry problem)
Example: Quad Meshing
The Main Problem in Parameterization

• You have a triangle mesh homeomorphic to a disc, possibly with holes
• You want to flatten this mesh into 2D with some specific properties
  • No foldovers, minimize distortion
The hard problem we will ignore for now

Not a Topological Disc
Projections

- Planar
- Cylindrical
- Spherical
Mass-Spring-Type Approaches

• Define energies/forces based on lengths and angles
  • Turned out to be the “Green-Lagrange” deformation tensor

• Integrate!

• Slow
• Not reliable
• ...

Interactive Texture Mapping, Maillot et al SIGGRAPH 93
Pelt Mapping
First problem to solve: no foldovers

• “Bijection”

• Triangle meshes are Graphs

• There is a sub-topic of Graph Theory called “Graph Drawing”, which is much older than Graphics...
Tutte Embedding

HOW TO DRAW A GRAPH

By W. T. TUTTE

[Received 22 May 1962]

1. Introduction

We use the definitions of (11). However, in deference to some recent attempts to unify the terminology of graph theory we replace the term ‘circuit’ by ‘polygon’, and ‘degree’ by ‘valency’.

A graph $G$ is 3-connected (nodally 3-connected) if it is simple and non-separable and satisfies the following condition; if $G$ is the union of two proper subgraphs $H$ and $K$ such that $H \cap K$ consists solely of two vertices $u$ and $v$, then one of $H$ and $K$ is a link-graph (arc-graph) with ends $u$ and $v$. 
Tutte Embedding

• You have a 3D network of nodes and edges
• Can you draw a 2D version of this network where no edges overlap?

• Define each vertex as convex combination of neighbours
  • Each vertex should lie at centroid of neighbours – “uniform” weights

• Fix open boundary to a convex shape (circle, square, polygon)
  • Must be convex or interior vertices can “escape” the boundary in 2D

• Solve system of linear equations
The formula

\[ \sum_{i=1}^{k} \lambda_i v_i = v_0 \]

Vertex \( v_0 \) is a linear combination of its neighbours \( v_i \). \( \lambda_i \) are the scalar “weights”.

Weights are constructed locally!
The formula

\[ \sum_{i=1}^{k} \lambda_i v_i = v_0 \]

Vertex \( v_0 \) is a linear combination of its neighbours \( v_i \).
\( \lambda_i \) are the scalar “weights”

\[ \sum_{i=1}^{k} \lambda_i = 1 \]

The weights sum to 1
The formula

\[ \sum_{i=1}^{k} \lambda_i v_i = v_0 \]

Vertex v_0 is a linear combination of its neighbours v_i. \( \lambda_i \) are the scalar “weights”

\[ \sum_{i=1}^{k} \lambda_i = 1 \]

The weights sum to 1

...If not: \[ \lambda_i = \frac{w_i}{\sum_{j=1}^{k} w_j} \]
The formula

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex $v_0$ is a linear combination of its neighbours $v_i$

$\lambda_i$ are the scalar “weights”

$$\sum_{i=1}^{k} \lambda_i = 1$$

The weights sum to 1

$\lambda_1, \ldots, \lambda_k \geq 0$

The weights are positive  

(often this rule is “bent”)

(If weights are not positive, $v_0$ can escape from one-ring)
The formula

\[ \sum_{i=1}^{k} \lambda_i v_i = v_0 \]

\[ \sum_{i=1}^{k} \lambda_i = 1 \]

\[ \lambda_1, \ldots, \lambda_k \geq 0 \]

Vertex \( v_0 \) is a linear combination of its neighbours \( v_i \). \( \lambda_i \) are the scalar “weights.”

The weights sum to 1

The weights are positive (often this rule is “bent”)

(If weights are not positive, \( v_0 \) can escape from one-ring)
Ok, Bijection. But...

- Uniform weights completely discard triangle shape information
- Even in 2D mesh, most vertices not at centroids of neighbours...
Shape-Preserving Weights

• Construct weights that would be correct for planar meshes

• Locally flatten each one-ring
  • “geodesic polar map”

• In this local 2D space, geometrically construct weights that reproduce $P$ exactly (see paper)
Harmonic Maps

- Weights based on edge lengths and areas
  - Can be negative, so parameterization can have fold-overs

- Map open boundary to Convex N-gon

- Solve sparse linear system

Multiresolution Analysis of Arbitrary Meshes, Eck et al, SIGGRAPH 95
Mean Value Weights

• Positive by construction

\[ w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{||v_i - v_0||} \]

• “Reproducing” / “Linear Precision” in planar mesh case

• Have been extended to general N-gons, 3D meshes, ...

\[ \tan(\theta) \geq 0 \text{ in } [0,90) \]
Parameterization with uniform weights [128] on a circular domain.

Parameterization with harmonic weights [28] on a circular domain.

Parameterization with mean value weights [33] on a circular domain.

Not a lot of flexibility once boundary is fixed...
“Virtual Boundary”
MIPS

• Re-cast existing work as energy minimization

1) find a parametrization for the boundary points, and
2) minimize an edge-based energy function

\[ E = \frac{1}{2} \sum_{\{i,j\} \in \text{Edges}} c_{ij} \| p_i - p_j \|^2 \]

to determine the parametrization for the inner points.

MIPS: An Efficient Global Parameterization Method, Hormann and Greiner 2000
MIPS

• Asked “What is the best-case scenario?”

• **Isometric** mapping – lengths and angles preserved

• Not possible for a 3D shape – but how close can we get?

• **Most Isometric ParameterizationS**
  • Basically, minimize a function of per-triangle deformation
  • iterative nonlinear optimization

MIPS: An Efficient Global Parameterization Method, Hormann and Greiner 2000
Distortion of Linear Map

\[ g(x) = Ax + b \]

\[ U^t A V = \Sigma = \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} \]

(Singular Value Decomposition)

“Metric Distortion”
MIPS

(Start with foldover-free convex-boundary embedding – eg Tutte)

repeat

choose a vertex $p_i$ by random

let $t_{i_1}, \ldots, t_{i_n}$ be the triangles that surround $p_i$

fix the positions of all parameter values except for $p_i$

minimize the local deformation energy

$$\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{ik})$$

in order to get the optimal position $\tilde{p}_i$ of $p_i$

until numerical convergence

$$\kappa_F(A) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}$$

$$= \frac{4E_D(g)}{\det \partial \psi}$$

$$= \frac{\text{trace}(A^t A)}{\det A}$$
MIPS

(Start with foldover-free convex-boundary embedding – eg Tutte)

repeat

choose a vertex $p_i$ by random

let $t_{i1}, \ldots, t_{in}$ be the triangles that surround $p_i$

fix the positions of all parameter values except for $p_i$

minimize the local deformation energy $\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{ik})$

in order to get the optimal position $\tilde{p}_i$ of $p_i$

until numerical convergence

$$\kappa_F(A) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}$$

$$= \frac{4E_D(g)}{\det \partial \psi}$$

Dirichlet Energy

Using Most Isometric Parametrizations for Remeshing Polygonal Surfaces, Labsik et al, 2000
Fig. 4. Data set and gray-coded Dirichlet energy of different parametrizations.
MIPS: An Efficient Global Parameterization Method

repeat

choose a vertex \( p_i \) by random

let \( t_{i_1}, \ldots, t_{i_n} \) be the triangles that surround \( p_i \)

fix the positions of all parameter values except for \( p_i \)

minimize the local deformation energy
\[
\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{i_k})
\]
in order to get the optimal position \( \tilde{p_i} \) of \( p_i \)

until numerical convergence
Stretch-Minimizing Parameterization

\[ L_2(T) = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}} \quad L_\infty(T) = \max(\sigma_1, \sigma_2) \]

- Iterative optimization starting from initial embedding
- Uses line searches – faster but requires fixed boundary (?)
- Not Smooth!!

Texture Mapping Progressive Meshes, Sander et al, SIGGRAPH 2001
Dirichlet Energy / Cotangent weights

\[ \frac{\partial E_A}{\partial u_i} = \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (u_i - u_j) = 0 \]

- Harmonic Map (Eck 94, Pinkall & Polthier 93)
- Can be negative! But, **Symmetric** (good for linear solvers)
- Gradient of 1-ring area wrt \( u_i \)
Isometric / Conformal / Authalic

• Isometric: Preserve Angles and Areas
  • Impossible

• Conformal: Preserve Angles
  • Harmonic

• Authalic: Preserve Areas
  • ???

Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002
Authalic Parameterization

• “Chi Energy”

\[
\frac{\partial E_{\chi}}{\partial u_i} = \sum_{j \in N(i)} \frac{(\cot \gamma_{ij} + \cot \delta_{ij})}{|x_i - x_j|^2} (u_i - u_j) = 0
\]

• Preserves relative areas of one-rings, “locally”

Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002
Natural Conformal Parameterization (DNCP)

• “the derivative of the Dirichlet energy on a triangle with respect to one of its vertices is equal to the opposite edge rotated by 90 degrees”

\[ \sum_{\Delta_{ijk}} \cot \alpha (u_i - u_j) + \cot \beta (u_i - u_k) = \sum_{\Delta_{ijk}} R_{ij}^{90} (u_k - u_j) \]

• Couples x and y!
• Still have to (arbitrarily) pin two vertices to fix rotation/translation

Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002
Dirichlet Energy

\[ E_D(u) = \frac{1}{2} u^t L_D u \]

Mesh Area

\[ A(u) = \frac{1}{2} u^t A u \]

Conformal Energy

\[ E_C(u) = \frac{1}{2} u^t L_C u \]

\[ L_C = L_D - A \]

By Definition: \( A(u) \leq E_D(u) \)
DNCP = LSCM

• Least Squares Conformal Maps published at same time as DNCP
• Math looks different, but ultimately $E_{LSCM}(u) = E_D(u) - A(u)$
• Both share problem: two arbitrary vertices must be pinned, introduces distortion:
Spectral Conformal Parameterization

- Instead of using a linear solver, use an Eigensolver
Machine Learning / Dimensionality Reduction

• Operate on points, not meshes

• LEM – Laplacian EigenMaps
  • Eigensolver applied to Dirichlet energy

• LLE – Locally Linear Embedding
  • Weights found by local minimization of covariance in planar projection
  • Minimize Bilaplacian w/ Eigensolver
Dimensionality Reduction Cont.

- **Maximum-Variance Unfolding**
  - Use Semi-definite programming to attempt to preserve all edge lengths

- **Hessian Eigenmaps**
  - LLE with local Hessians

- **LTSA: Local Tangent-Space Alignment**
  - Flatten local regions separately, then solve stitching problem
Weird Stuff

Free-boundary Tutte Paramaterization

“Parameterization Repair” – use uniform weights where cotan < 0
Conformal: Angle-Based Flattening

• Iterative nonlinear optimization to explicitly preserve one-ring angles
  • Sum of ring angles = 2π, sum of triangle angles = π
  • “Wheel consistency” — one-rings match up

• Provable convergence, but slow

• ABF++: faster, but still iterative solve
• LABF: linearized, “less” conformal but ~10x faster

• Even the Linear variant handles some situations DNCP fails on
  • Because it does not have the negative-weights problem

Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening, Sheffer and de Sturler, 2001
Circle patterns

• Conformal == Preserves Circles and circle intersections

• Place circle at each face
• They intersect at vertices
• Solve for radii of 2D circles that preserve angles

• Hard.

Discrete Conformal Parameterizations via Circle Patterns, Kharevych et al, TOG 2006
Cone Singularities

- Imagine you could just “redistribute” curvature between vertices
- “Push” all curvature to a few vertices, so the rest have 0 curvature
  - So mesh is Developable except at a few points
- Cut mesh between those points
- Now entire mesh is developable!
- If done properly, parameterization along cuts perfectly lines up
Cone Singularities
Circle Patterns w/ Cone Singularities
Circle Patterns w/ Cone Singularities

Continuous across cuts!!!

Conformal Flattening by Curvature Prescription and Metric Scaling, Ben-Chen et al, Eurographics 2008
Discrete Conformal Parameterizations via Circle Patterns, Kharevych et al, TOG 2006
Conformal Equivalence

• DDG formulation of “conformal equivalence”
• Convex energy, efficient solution
  • Hessian is cot-Laplace op, of course
• Cone singularities are automatic
• Desired distortion can be prescribed
• They win. Global Conformal Parameterization solved.

* Texture has to be periodic

Conformal Equivalence of Triangle Meshes, Springborn et al, SIGGRAPH 2008
ARAP: As Rigid As Possible

- Start from initial flattening (DNCP, Shape-Preserving)

- Iterate:
  - Solve for separate optimal rotation of each triangle
  - Solve for vertex positions that best satisfy rotations and triangle shapes (global Poisson problem)

- “Unwrap” tool in Meshmixer

Spherical Parameterization

- Many adaptations: Spherical ABF, Conformal, ARAP, ...
Toroidal Parameterization
Polycube Maps
Seams / Charting
Global Bijections
Local Parameterization...
Local Parameterization
Ryan Schmidt
Decal Parameterization Problem
Global Parameterization
(Appplied Locally)
Global Parameterization

(Applied Locally)
Global Parameterization
(Applied Locally)
Desirable Properties

• **Predictable** behavior
  • Distortion is consistent
  • Bounded UV-space

• Fast

• Smooth

Usually!

Yes!

Sometimes!
Discrete Exponential Map

- Assume isometric maps exist between tangent spaces $T_x$
  - Only true on Sphere

- Then transformation between tangent spaces is affine

- Incrementally lift neighbours of $p$ into $T_p$ via tangent-space vector summation
Discrete Exponential Map

• Input is set of points and normals
  • Does not require mesh, just neighbours

• Propagate parameterization $U$ from seed point $p$

• For point $x$ on propagation front
  1) Project $(x-q)$ into tangent plane at $q$
  2) Apply rotation $R$ that takes frame at $q$ to frame at $p$
  3) Add to $U(q)$ to get $U(x)$
"Vector Dijkstra"
Discrete Exponential Map

[SIGGRAPH 2006]
Discrete Exponential Map

[SIGGRAPH 2006]
Discrete Exponential Map

[SIGGRAPH 2006]
Discrete Exponential Map
Bad meshes were a problem...

One bad projection ruins everything “downwind”
Improved Discrete ExpMap

1) Average uv-predictions from multiple upwind samples

2) Use smoothed normal field
“moving planar projection”
Lots of applications...

• ~100 citations
• Very useful in 3D sculpting/painting
  • Basis for meshmixer
• Local geometry editing
  • Low-distortion 2D domain for stitching/etc
• Local Geodesics
  • Surface signal analysis, descriptors, ...

• Improved by Campen et al
  • More robust but more complicated to implement
Parallel Transport

“Apply rotation $R$ that takes frame at $q$ to frame at $p$”
Frame Propagation

• Frame is \((e_1, e_2, n)\)
• To compute frame at \(x\):

\[
e_1(x) = \sum_{q_i \in N_u(x)} w(q_i, x)R(q_i, x)e_1(q_i)
\]

• \(R\) is rotation around \((n_q \times n_x)\) that takes \(n_q\) to \(n_x\)
  • Levi-Civita Connection

• \(w\) is inverse-distance weight
• Find \(e_2(x)\) by orthogonality
Stroke Parameterization
$d_x$ is a **signed** distance (see paper)
Frames at $p_i$ are Darboux frames $(S', S' \times n, n)$
Applications
Analysis
Geodesic Error
FP vs Conformal Map
FP vs ARAP
Limitations: Self-Intersection

• Not clear that there is a right answer...
• Still not solved in 2D, either

By-Example Synthesis of Curvilinear Structured Patterns, Eurographics 2013
(Next paper session in Room B.1)
Limitations: Distortion / Smoothness
Limitations: Overlap Discrimination
Future Work

• Better constraints for ARAP

• Improved overlap handling

• Closed Loops
  • map to cylinder