Mesh Editing with Poisson based Gradient Field Manipulation

Yizhou Yu[1]
Kun Zhou[2]
Dong Xu[2,3]
Xiaohan Shi[2,3]
Hujun Bao[3]
Baining Guo[2]
Heung-Yeung Shum[2]

[1]University of Illinois at Urbana Champaign
[2]Microsoft Research Asia
[3]Zhejiang University

Presented by: Gagan Bansal

Basic Technique

- Modify the vertex positions of the mesh by manipulating the gradient field
  - Modify the gradient field of the mesh
  - Apply the boundary conditions
  - Reconstruct the mesh by solving the Poisson Equation

Poisson Equation

$\Delta f = \mathrm{div}(w)$, f is the unknown & w is the guidance field

Triangle Mesh – Not a regular Grid
Approximate by discrete field and redefine divergence for discrete field

Discrete vector field for mesh – Piecewise constant function with domain being vertices of the mesh
- Constant coplanar vector defined for each triangle

Helmholtz Hodge vector field decomposition

$$w = \nabla(\phi) + \nabla(v) + h$$

Φ is scalar potential field
V is vector field $\Delta v = 0$
h is some field which is curl and divergence free

Φ happens to be solution of least square minimization $\min_{\phi} \int \int_{\Omega} \|\nabla \phi - \mathbf{w}\|^2 dA,$

Poisson Equation

Discrete potential field - $\phi(x) = \sum_i B_i(x)\ \phi_i$, B_i is 1 only at v_i

Divergence

$$(\text{Div } w)(v_i) = \sum_{Tk \text{ in } N(i)} \nabla (B_{ik}).\ w\ |T_k|$$

N(i) is set of triangles incident on v_i

$|T_k|$ is area of Triangle T_k

Grad(B_{ik}) is gradient vector of B_i within T_k

$$\text{Div } (\nabla (\phi)) = \text{Div } w$$

Which is a sparse linear system

$$A\ f = b$$

Can be solved using Conjugate Gradients

Poisson Mesh Solver

- Apply Poisson equation to solve unknown target mesh with known vertex connectivity but unknown vertex coordinates
- A f = b
  - b calculated from divergence values at all vertices
  - A independent of guidance field
- Solve thrice once for each coordinate x ,y ,z
- Guidance vectors associated with larger triangles in source mesh are better approximated hence weight factor of $|T_k|$

Gradient Field Editing using Local Transforms

- Local Transformation can be defined for each triangle
- New Gradient field can be defined for all the three coordinates
- Guidance field → Use Poisson as earlier to get the new mesh

Boundary Condition Editing

- Local or global mesh editing by manipulating features like curves and vertices.
- Boundary condition of a mesh (BC) – (I, P, F, S, R)
  - I is index set of connected vertices of the mesh
  - P is 3d position of mesh vertices
  - F is set of local frames which define local orientation of vertices (3 orthogonal vectors)
  - S is scaling factor associated with the vertices
  - R is strength field (min distance between free vertex and constrained vertex)
- Vertex is free if it does not belong to any BC else it is constrained

Boundary Condition Editing

- After editing BC’ = (I, P’, F’, S’, R)
- Propagate the local frame and scale changes from constrained vertices to free vertices
- Assign free vertex (V_f) the weighted average of local frame and scale changes of all constrained vertices V_c
- Uniform weighing scheme
  - All constrained vertices are weighed equally
- Linear weighing scheme
  - Weight proportional to inverse of distance between free and constrained vertex
- Gaussian weighing scheme
  - Weight given as $\exp\left(-\frac{(dist(\mathbf{v}_f,\mathbf{v}_c)-dist(\mathbf{v}_f,\mathbf{v}_{min}))^2}{2\sigma_d^2}\right)$
  - V_{min} is constrained vertex closest to V_f

Boundary Condition Editing

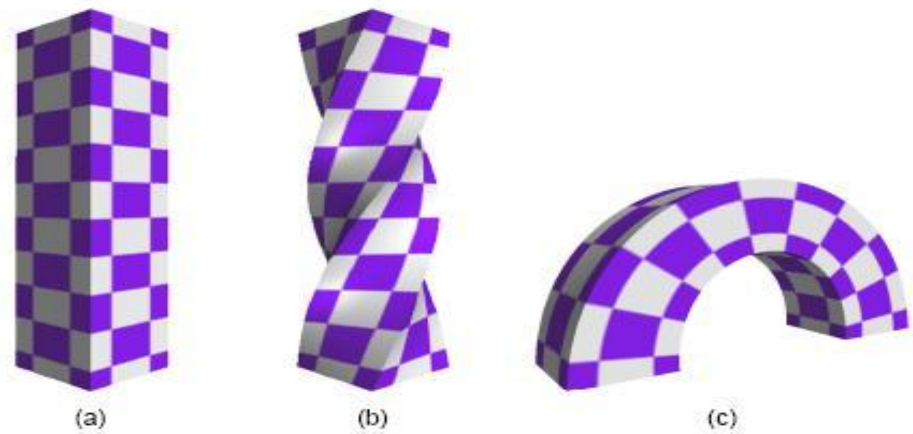

Figure 4: (a) Original model (2040 vertices and 4000 faces), (b) twisting by rotating the top rectangular boundary around the vertical axis of the PRISM (running time = 578 ms), (c) bending by rotating the top boundary around a horizontal axis in addition to a translation (running time = 609 ms).

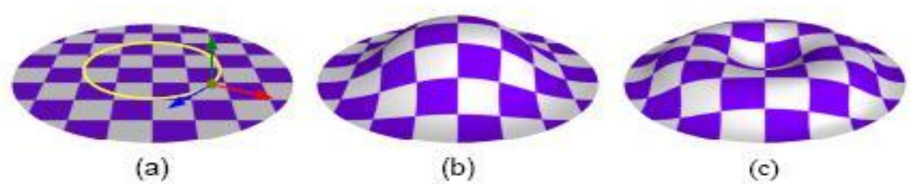

Figure 5: (a) Original model (1281 vertices and 2480 faces), (b)-(c) simultaneous normal rotation around their respective tangents using cosine functions with two different phase angles as their strength fields. The running time for (b) is 230ms and (c) 240ms.

Applications

- Mesh Deformation
- Mesh Merging
- Mesh Smoothing and Denoising

Mesh Deformation

- Curves or vertices as BC
- Fixed BC – vertices fixed
- Editable BC- vertices to be modified
- Rest free vertices
- Vertices of same editable curve can be modified individually or simultaneously(same transformation to each vertex)

Mesh Deformation

Figure 6: Interactive mesh deformation. The top row, from left to right, shows the original model and the result from rotating normals of a curve around their respective tangents. The bottom row shows the results by applying a translation or rotation to the whole curve. The curve is around the neck, and the weighting scheme for all the constrained vertices on the curve is Gaussian.

Mesh Deformation SpeedUp

- $Af=b$ --------- > $f=A^{-1}b$
- A^{-1} fixed and hence can be computed by LU decomposition
- Multiresolution mesh, Poisson editing only at coarsest level

Mesh Merging

- Vertex correspondence at boundaries of both the meshes
- Compute local frames along two boundaries
- Obtain intermediate boundary including both vertex positions and local frames by either choosing any of the mesh boundaries or interpolating the two mesh boundaries.
- Compare local frames at intermediate boundary with those at original boundaries and obtain the 3d rotations.
- Propagate the rotations in both the meshes
- For all the vertices in both the meshes, solve using Poisson.

Mesh Merging

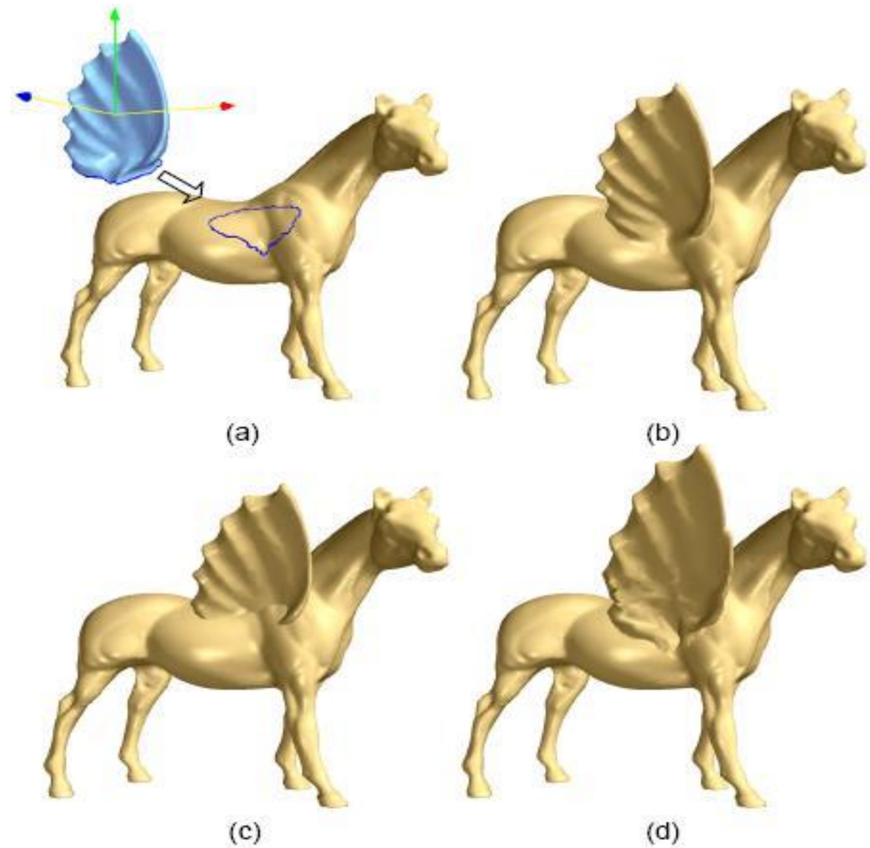

Figure 9: (a) the boundary on the WING (2000 faces) is projected along a user-defined direction to define the boundary on the HORSE model (100K faces), (b) the result from our projection scheme (running time = 400 ms), (c) Boolean operation, (d) WIRE.

Mesh Merging

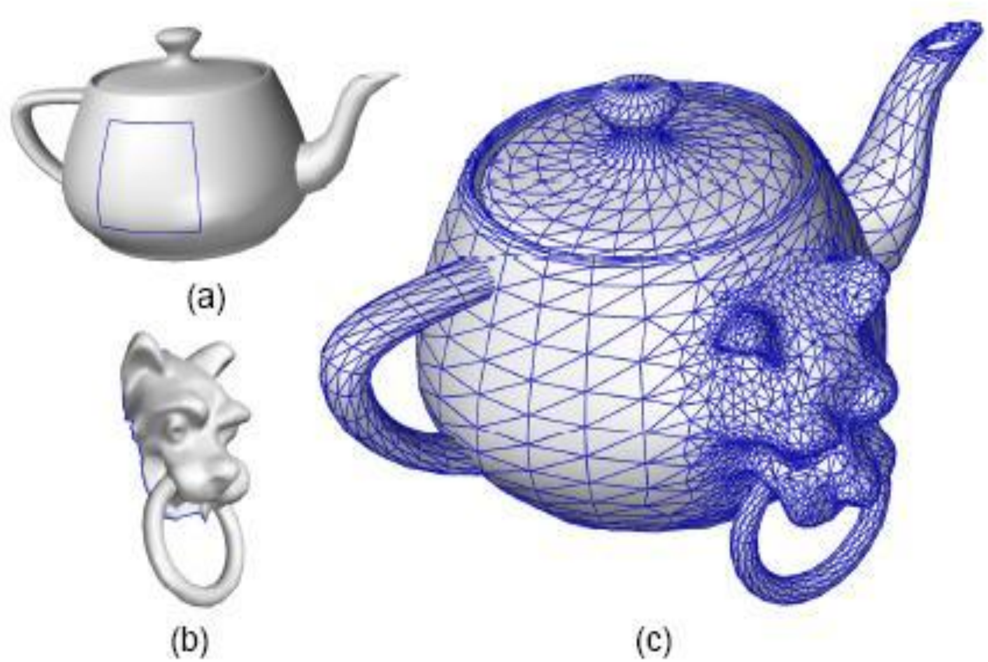

Figure 11: Object merging by interactively specifying sparse key vertex correspondences between two boundaries. GARGOYLE has 4000 faces, TEAPOT has 2000 faces, and the running time is 890 ms.

Mesh Smoothing and Denoising

- Original normal n_i and new normal n_i'
- Local rotation matrix is obtained and hence a guidance field is obtained.
- Can be solved as a linear system as before

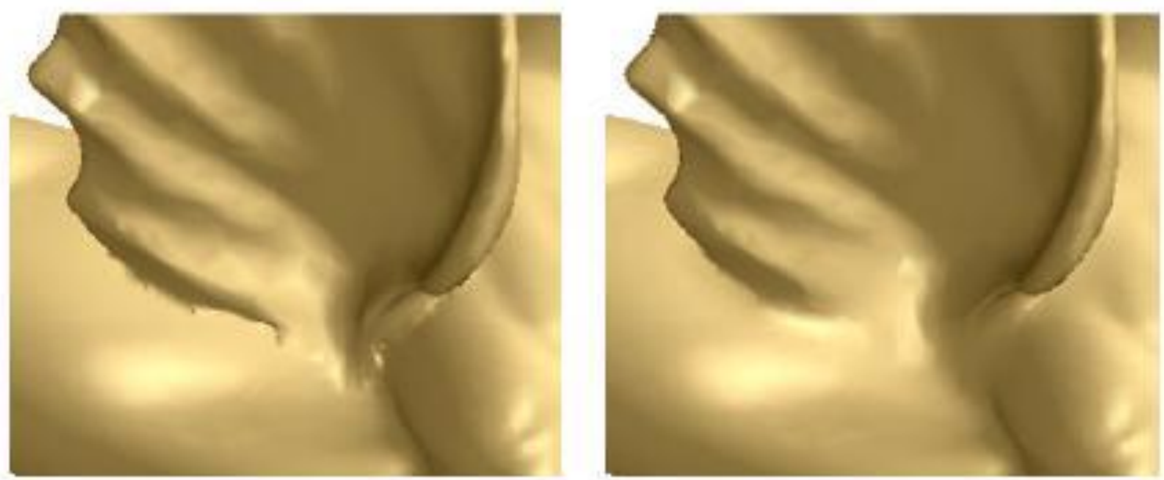

Figure 14: Smoothing merging boundary. Left: before smoothing, Right: after smoothing.