Computing Surface PolyCube-Maps by Constrained Voxelization

Yang Yang, Xiao-Ming Fu, Ligang Liu
University of Science and Technology of China
Motivation
PolyCube:
1. Compact representations for closed complex shapes
2. Face normal aligns to the axes
3. Axes: $\pm (1,0,0)^T$, $(0, \pm 1,0)^T$, $(0,0, \pm 1)^T$

PolyCube-Map $f$:
1. A mesh-based map
2. Low isometric distortion
3. No flipped triangles
Applications for PolyCube-Map:

1. Texture mapping
2. GPU-based subdivision
3. All-Hex mesh generation
Requirements on PolyCube-Maps

1. Bijective

2. The mapping exhibits as little isometric distortion as possible

3. The PolyCube contains as few corners as possible
Related work – PolyCube-Map Computation

Deformation-based methods

Invalid Polycubes
May not accurately represent the similarity

Not sensitive to flipped triangles

Projection-based methods

High Distortion

Construction-based method

[Gregson et al. 2011] [Fu et al. 2016] [Livesu et al. 2013] [Huang et al. 2014]

[Tarini et al. 2004] [Lin et al. 2008] [He et al. 2009] [Yu et al. 2014]
Contribution

• A novel method to compute surface PolyCube-maps with low isometric distortion

• Key components:
  • A robust PolyCube construction method based on an erasing-and-filling strategy
  • A novel surface mapping technique based on a use of the quad mesh optimization algorithm
Algorithm
Algorithm workflow

Input:
A source mesh & a pre-axis-aligned shape

Output:
PolyCube-map

I. Constrained voxelization
II. Computing surface PolyCube-Map
Constrained voxelization: Formulation

\[
\begin{align*}
\min_{C} & \quad \mathcal{N}(C) \\
\text{s.t.} & \quad d_h(C, A) \leq K_h, \\
C & \in \mathcal{T}
\end{align*}
\]

- \( \mathcal{N}(C) \): The number of corners of the PolyCube
- \( d_h(C, A) \leq K_h \): The error-bounded constraint
- \( C \in \mathcal{T} \): The topological constraint
Two operators

Performing one filling operator:
An erasing operator performed on the dual domain

Performing one erasing operator:
1. Partition the voxels domain into disjoint cuboids
2. Compute the corner number changes
3. Remove the cuboid that causes the most changes
Erasing-and-filling Strategy

1. Generate the initialization, $k = 1$
2. Perform one erasing operator without violating the constraints
3. Perform one filling operator without violating the constraints
4. $k = k + 1$, go to step 2
Our developed erasing and filling operators are effective and robust.
Algorithm workflow

I. Constrained voxelization

Input:
A source mesh & a pre-axis-aligned shape

II. Computing surface PolyCube-Map

Output:
PolyCube-map

Pre-axis-aligned shape $A$
Constructed PolyCube $C$
Optimized quad mesh $Q$
Segmentation $S$
PolyCube-map $f$
PolyCube-Maps computation

1. Optimize a new quad mesh

2. Segment the triangular surface into a set of submeshes

3. Map each submesh onto one PolyCube chart
Quad mesh optimization

Requirements

1. Preserve the given shape with the same connectivity of the Polycube
2. All edge lengths are equal to a constant
3. All interior angles are in an interval $[\frac{\pi}{2} - \theta, \frac{\pi}{2} + \theta]$ 
4. Almost no flipped quads

Anderson acceleration [Peng et al. 2018]
Segmentation and map computation

PolyCube → Quad mesh → Triangular surface

Same connectivity  Shape preserving

Decompose
2D fixed-boundary mapping

AMIPS method [Fu et al. 2016]

chart
submesh
submesh
Experiments

Quality measure – isometric distortion:

\[ \delta^{iso} = \max \left\{ \sigma_{max}, \frac{1}{\sigma_{min}} \right\}, \delta^{iso}_{opt} = 1 \]
Mixed objective

\[ \min_C N(C) + \beta d_h(C, \mathcal{A}) \]
\[ \text{s.t. } C \in \mathcal{T} \]

- \( \beta = 1 \) 
  
  \( N(C) = 79, \ d_{avg} = 2.53 \)

- \( \beta = 15 \) 
  
  \( N(C) = 153, \ d_{avg} = 1.31 \)

- \text{Ours} 
  
  \( N(C) = 155, \ d_{avg} = 1.27 \)

1. Selecting a fixed weight for all models is \text{non-trivial}
2. Improper weights may result in \text{large} isometric distortion
Voxel sizes

More iterations of the erasing-and-filling solver are needed for smaller voxel sizes with similar results.
Quad mesh optimization

Benefits of meeting the optimization requirements

Input

Laplacian smoothing

\( N(C) = 65, d_{avg} = 1.32 \)

Ours

\( N(C) = 65, d_{avg} = 1.19 \)
Various pre-axis-aligned shapes

The final PolyCubes are closely related to the input pre-axis-aligned shapes

[GSZ11]  
\(N(C) = 70, d_{avg} = 1.30\)

[HJS*14]  
\(N(C) = 57, d_{avg} = 1.33\)

[FBL16]  
\(N(C) = 76, d_{avg} = 1.21\)
Different triangulations

Isotropic\((N(C) = 76, d_{avg} = 1.33)\)

Sparse\((N(C) = 72, d_{avg} = 1.35)\)

Anisotropic\((N(C) = 72, d_{avg} = 1.34)\)

Dense\((N(C) = 72, d_{avg} = 1.35)\)

Almost the same number of corners
Different Hausdorff distance bounds

$K_h = 0.5\% \ l_b$

$(N(C) = 102, \ d_{avg} = 1.17)$

$K_h = 1.5\% \ l_b$

$(N(C) = 78, \ d_{avg} = 1.23)$

$K_h = 3.5\% \ l_b$

$(N(C) = 44, \ d_{avg} = 1.26)$

Smaller $K_h$, more corners of the PolyCube, lower distortion of the PolyCube-map
Data set: 300 examples
Data set: All-hex meshes
Comparison to [Fu et al. 2016]

- **Ours**
  - $(N(C) = 67, d_{avg} = 1.29)$

- [FBL16]
  - $(N(C) = 192, d_{avg} = 1.65)$
  - $(N(C) = 128, d_{avg} = 1.38)$

39 failures
Comparison to [Yu et al. 2014]

\[ N(C) = 82, \ d_{avg} = 1.31 \]  

\[ N(C) = 72, \ d_{avg} = 1.26 \]  

\[ N(C) = 82, \ d_{avg} = 1.29 \]  

\[ N(C) = 55, \ d_{avg} = 1.24 \]
Comparison to [Huang et al. 2014]

(Huang et al. 2014) *14

\(N(C) = 68, 36 \text{ flipped triangles})

\(N(C) = 56, 81 \text{ flipped triangles})

(Huang et al. 2014) *14

\(N(C) = 42, d_{avg} = 1.31\)

\(N(C) = 74, d_{avg} = 1.35\)

[HJS*14]

Ours
Comparison to [Livesu et al. 2013]

\[ N(C) = 172, \ 13 \text{ flipped triangles} \]

\[ N(C) = 179, \ d_{avg} = 1.24 \]

\[ N(C) = 94, \ 2 \text{ flipped triangles} \]

\[ N(C) = 97, \ d_{avg} = 1.34 \]
Conclusions

• A novel approach to compute surface PolyCube-maps with low isometric distortion

• Limitations:
  • Global overlaps
  • Sharp features
  • Theoretical guarantee
Thank you!

Code and data are available at: http://staff.ustc.edu.cn/~fuxm/