Cubic stylization

Team 04

ARYA SINHA and SHANTANU DIXIT

1 INTRODUCTION

There is a shortage of tools for creating artistically styled 3D geometry, which can be a significant hurdle for those interested in geometric stylization.

The focus in this paper is on the unique style of cubic sculptures, which have made a mark in the art world thanks to artists like Pablo Picasso and are commonly seen in digital animations, as well as practical applications like furniture and architectural design. The paper provides a method to take a 3D shape as an input and produces a deformed shape, mimicking the distinctive cubic style.

The approach to cubic stylization formulates the task as an energy optimization. This is done to maintain the geometric details while giving the shape a cubic transformation. This is achieved this by combining an 'as-rigid-as-possible' (ARAP) energy to preserve local structure with an $\mathcal{L}_1$ norm of the rotated normal to encourage a cube-like appearance. $\mathcal{L}_1$ norm has been chosen because it encourages sparsity. To optimize this energy, standard local-global update strategy has been used.

2 LITERATURE REVIEW

(1) Legolization [4] as a type of stylization in the form of LEGO bricks, creates a stable layout for LEGO bricks based on an input 3D model. Unlike other methods that use heuristic-based metrics, it uses force-based metric. It uses a method that uses the forces acting on the sculpture to determine how stable it is. Their approach does two things: it tells which structure is more stable and ranks them, and it also sets a limit for what's considered stable.
(2) Liu and Jacobson's [3] presented stylization as a shape analogy problem, with surface normals as the defining relationship. They introduced spherical shape analogies $A : A' :: B : B'$, where one element A of the analogy is a unit sphere, allowing for versatile operations on input mesh B. This constrained approach, although limited to using a sphere as A, proved effective in achieving various geometric styles. However, this method is designed for transfering styles, instead of generating 3D stylized objects directly.
(3) Sorkine and Alexa [6] presented a mesh deformation technique known as ARAP (As-Rigid-As-Possible), ensuring intact mesh details during shape changes. The approach involved maintaining the length through rigid transformations, avoiding stretching, flattening, or shearing. The implementation of ARAP involves the use of rotation matrices to allow some flexibility in the deformation, accommodating rotations as seen in animations. The deformation algorithm aims to minimize deviations from these conditions in a least-squares sense, offering an As-Rigid-As-Possible transformation for all edges. An Alternating Minimization approach is employed to find optimal solutions for both vertex positions and rotation matrices, resulting in a natural and visually pleasing deformation.
(4) Manson and Schaefer [5] introduced a deformation method for 3D models capable of dealing with complex models containing millions of vertices. The approach is to perform a deformation on a simplified, low-resolution version of the model and then add the details back while staying within certain constraints. This ensures the overall shape of the model is maintained, providing as rigid as possible deformations [6].
(5) A polycubic technique including the minimization of the $\mathcal{L}_1$ norm of normals on the deformed tetrahedral mesh with ARAP for regularization was proposed by Huang et al [1]. A similar methodology was used

Authors' address: Arya Sinha, arya20498@iiitd.ac.in; Shantanu Dixit, shantanu20118@iiitd.ac.in.

in this paper, with the important difference being that the $\mathcal{L}_1$ norm was defined on the original mesh's rotated normals.

3 MILESTONES

Table 1. Milestones

S. No.	Milestone	Member
	Mid evaluation	
1	Gain an understanding of different optimization precursors for ADMM optimization.	Shantanu
2	Gain an understanding of the Alternating Direction Method of Multipliers (ADMM) for the local update step.	
3	Understanding Boyd penalty parameter for better convergence.	Arya
4	Implement the proposed algorithm using OpenGL for the local update step.	
	Final evaluation	
5	Understand As-Rigid-As-Possible (ARAP) for Global step.	Shantanu
6	Implement the proposed algorithm using OpenGL for the global update step.	
7	Generate convergence plots to analyze the algorithm's performance. • Experiment with various stopping criteria. • Experiment with various cubeness degrees. • Experiment with various meshes.	Arya
8	Explore non-uniform cubic deformation by varying the lambda parameter.	

3.1 Milestones Achieved (till mid-evaluation)

- Read and understood method of multipliers for a better understanding of ADMM
- Read and understood Boyd's penalty parameter for local step
- Read and understood ADMM and scaled ADMM used in the paper
- Implemented local step

3.2 Challenges Faced

- New to optimization concepts
- Had difficulty in finding a starting point
- Had a hard time absorbing the mathematics intensive content of the paper

4 METHOD OVERVIEW

4.1 Alternative Direction of Method Multipliers (ADMM)

ADMM (Alternative Direction of Method Multipliers) is an optimization algorithm which solves problems of the form:

$$\begin{aligned} \text{minimize} \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = c, \end{aligned} \tag{1}$$

with variables $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and matrices $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, and vector $c \in \mathbb{R}^p$. The functions f and g are assumed to be convex.

The augmented Lagrangian is defined as:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2, \tag{2}$$

where y is the dual variable and $\rho > 0$ is the penalty.

The updates for x, z, and y in the ADMM algorithm are performed sequentially as follows:

$$x^{k+1} = \text{argmin}_x\ L_\rho(x, z^k, y^k), \tag{3}$$
$$z^{k+1} = \text{argmin}_z\ L_\rho(x^{k+1}, z, y^k), \tag{4}$$
$$y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \tag{5}$$

Liu and Jacobson [2] uses a scaled version of ADMM with the following updates:

$$x^{k+1} = \text{argmin}_x \left(f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|_2^2\right), \tag{6}$$
$$z^{k+1} = \text{argmin}_z \left(g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2\right), \tag{7}$$
$$u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c. \tag{8}$$

where $u = \frac{y}{\rho}$ is the scaled dual variable.

4.2 Algorithm

The method involves taking a triangle mesh as input an outputting a cubified shape having each sub component as an axis aligned cube. The first term in the below equation corresponds to ARAP [6] while the second preserves cubeness by minimizing the $\mathcal{L}_1$ norm of the rotated normal.

$$\sum_{i\in V}\sum_{j\in N(i)} w_{ij}^2\|R_i\mathbf{d}_{ij} - \mathbf{e}_{ij}\|_F^2 + \lambda a_i\|R_i\mathbf{n}_i\|_1 \tag{9}$$

For minimizing the above expression (optimizing energy) we follow a local-global update strategy. Local step finds the rotation matrix using scaled ADMM. The global step as in [6] updates the vertex positions based on the rotation matrix obtained from the local step. Here we discuss the local step as proposed by the authors.

4.3 Local Step

Algorithm 1 Local-Step-Cubic-Stylization

Require: Mesh Data
Ensure: Deformed vertex positions
Initialize:
$\tilde{V} \leftarrow V$
$R^* \leftarrow$ Rotation matrix
$z^k \leftarrow$ Rotated per vertex normal
$\rho^k \leftarrow$ Initial penalty parameter
$u^k \leftarrow$ Scaled dual variable
for each vertex i **do**
while not converged **do**
Compute $M_i = \begin{bmatrix} D_i & n_i \end{bmatrix} \begin{bmatrix} W_i & \mathbf{0} \\ \mathbf{0} & \rho^k \end{bmatrix} \begin{bmatrix} \widetilde{D_i}^T \\ (z^k - u^k) \end{bmatrix}$
Compute SVD $M_i = U_i \Sigma_i V_i^T$
Compute $R_i^{k+1} = V_i U_i^T$
Update z^{k+1} using shrinkage step:
$z^{k+1} \leftarrow S_{\lambda_a/\rho^k}(R_i^{k+1} \cdot n_i + u^k)$
Update u^{k+1}:
$u^{k+1} \leftarrow u^k + R_i^{k+1} \cdot n_i - z^{k+1}$
Update ρ^{k+1} based on Boyd
end while
end for
Ensure: $det(R_i^*)$ for all vertices > 0

REFERENCES

[1] Jin Huang, Tengfei Jiang, Zeyun Shi, Yiying Tong, Hujun Bao, and Mathieu Desbrun. 2014. 1-based construction of polycube maps from complex shapes. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 1–11.

[2] Hsueh-Ti Derek liu and Alec Jacobson. 2019. Cubic stylization. *ACM Transactions on Graphics* 38, 6 (nov 8 2019), 1–10. [Online; accessed 2023-10-20].

[3] Hsueh-Ti Derek Liu and Alec Jacobson. 2021. Normal-Driven Spherical Shape Analogies. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 45–55.

[4] Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. 2015. Legolization. *ACM Transactions on Graphics* 34, 6 (nov 2 2015), 1–12. [Online; accessed 2023-10-20].

[5] Josiah Manson and Scott Schaefer. 2011. Hierarchical deformation of locally rigid meshes. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 2387–2396.

[6] Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, Vol. 4. Citeseer, 109–116.