Cubic Stylization

Report Paper Code

Arya Sinha Shantanu Dixit

Team 04

Introduction

- The focus in this paper is on the unique style of cubic sculptures - cubic stylization

- Formulates the problem as an energy optimization process

$$\underset{\tilde{V}, \{R_i\}}{\text{minimize}} \sum_{i \in V} \sum_{j \in \mathcal{N}(i)} \underbrace{\frac{w_{ij}}{2} \|R_i d_{ij} - \tilde{d}_{ij}\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \|R_i \hat{n}_i\|_1}_{\text{CUBENESS}} .$$

- Main objective is to obtain a cubic transformation while maintaining the geometric details of the shape.

Overview

- Pre-requisites
  - ADMM - Alternating Direction of method of multipliers
- Methodology
  - Local Update Step
  - Global Update Step (As rigid as possible - ARAP)
- Results
  - Cubic stylized output
  - Non uniform cubic deformation
  - Convergence plots with varying
    - Stopping criteria
    - Cubeness degree

Alternating Direction of Method Multipliers (ADMM)

- Solves problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c, \end{array}$$

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

- Augmented Lagrangian is defined as

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

- Updates for x, z and y in the ADMM algorithm are performed sequentially as follows.

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, z^k, y^k), \\ z^{k+1} &= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k), \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \end{aligned}$$

- Paper uses a scaled version of the ADMM with following updates

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x \left(f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|_2^2\right), \\ z^{k+1} &= \operatorname{argmin}_z \left(g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2\right), \\ u^{k+1} &= u^k + Ax^{k+1} + Bz^{k+1} - c. \end{aligned}$$

Methodology

- Method involves taking a triangle mesh as input and outputs a cubified shape having each sub component as an axis aligned component.
- Authors formulate this as an energy minimization problem

$$\underset{\tilde{V},\{R_i\}}{\text{minimize}} \sum_{i \in V} \sum_{j \in N(i)} \underbrace{\frac{w_{ij}}{2} \|R_i d_{ij} - \tilde{d}_{ij}\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \|R_i \hat{n}_i\|_1}_{\text{CUBENESS}} .$$

- Minimizing the above expression (optimizing energy) involves a local-global update strategy
- Local step: Finds Rotation matrix using scaled ADMM
- Global step: Updates vertex positions based on obtained rotation matrix (using ARAP-RHS)

Local Update Step

Algorithm 1 Local-Step-Cubic-Stylization

Require: Mesh Data
Ensure: Deformed vertex positions

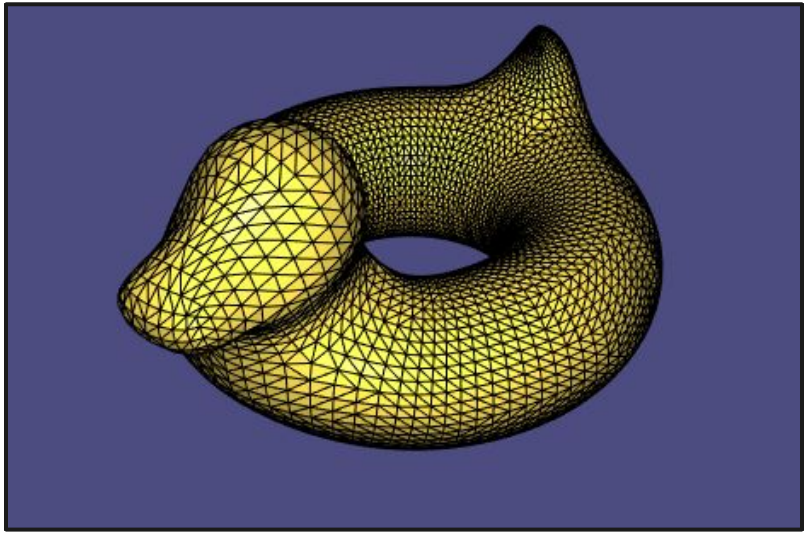
Initialize:
$\tilde{V} \leftarrow V$
$R^* \leftarrow$ Rotation matrix
$z^k \leftarrow$ Rotated per vertex normal
$\rho^k \leftarrow$ Initial penalty parameter
$u^k \leftarrow$ Scaled dual variable
for each vertex i **do**
  while not converged **do**
    Compute $M_i = \begin{bmatrix} D_i & n_i \end{bmatrix} \begin{bmatrix} W_i & \mathbf{0} \\ \mathbf{0} & \rho^k \end{bmatrix} \begin{bmatrix} \widetilde{D_i}^T \\ (z^k - u^k) \end{bmatrix}$
    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

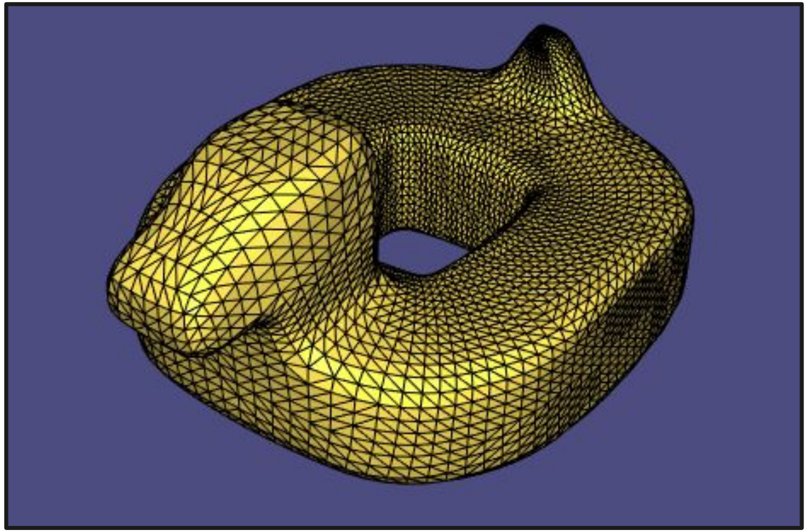
Global Update Step (ARAP)

Algorithm 2 As rigid as Possible - (Sorkine-Alexa)

Require: Mesh Data
Ensure: Deformed vertex positions
  Initialize:
  $V_{\text{original}} \leftarrow$ Original vertex positions
  $W \leftarrow$ Edge Weight Matrix
  $R_{\text{local}} \leftarrow$ Rotation matrix obtained from local step
  for each iteration **do**
    As-rigid-as-possible RHS
    Initialize RHS matrix: $RHSMatrix \leftarrow \mathbb{R}^{\text{number of vertices} \times 3}$
    for each vertex i **do**
      $rhs\_vertex\_sum \leftarrow \mathbb{R}^3$
      for each adjacent vertex j **do**
        $coeff \leftarrow w_{i,j}$
        $rows\_diff \leftarrow (V_{\text{original}}(i) - V_{\text{original}}(j))^T$
        $rot\_comb \leftarrow R_{\text{local}}[i] + R_{\text{local}}[j]$
        $product \leftarrow rot\_comb \cdot rows\_diff$
        $rhs\_vertex\_sum \leftarrow rhs\_vertex\_sum + coeff \cdot product$
      end for
      Update RHS matrix row: $RHSMatrix[i, :] \leftarrow rhs\_vertex\_sum$
    end for
  end for

Cubic stylized output

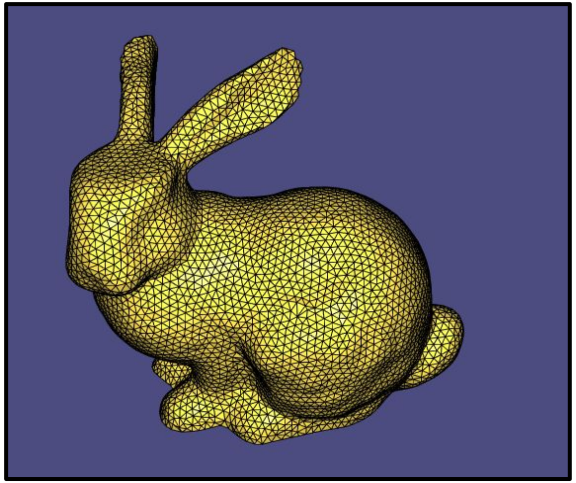




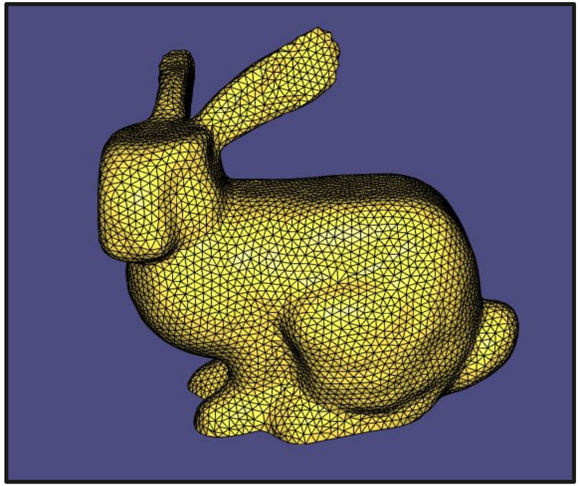
Cubeness degree = 0.8

Non uniform cubeness

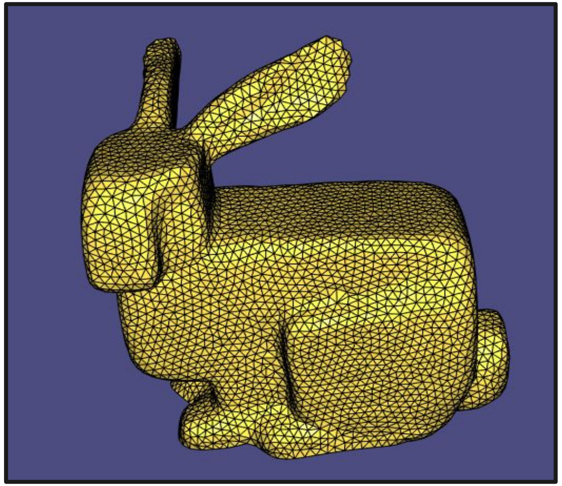
We vary λ across the surface to have different cubeness for different parts. We apply higher λ at the top of the object than the bottom.



At the bottom 75%, $\lambda = 0$, and it increases gradually to 0.8 to the top

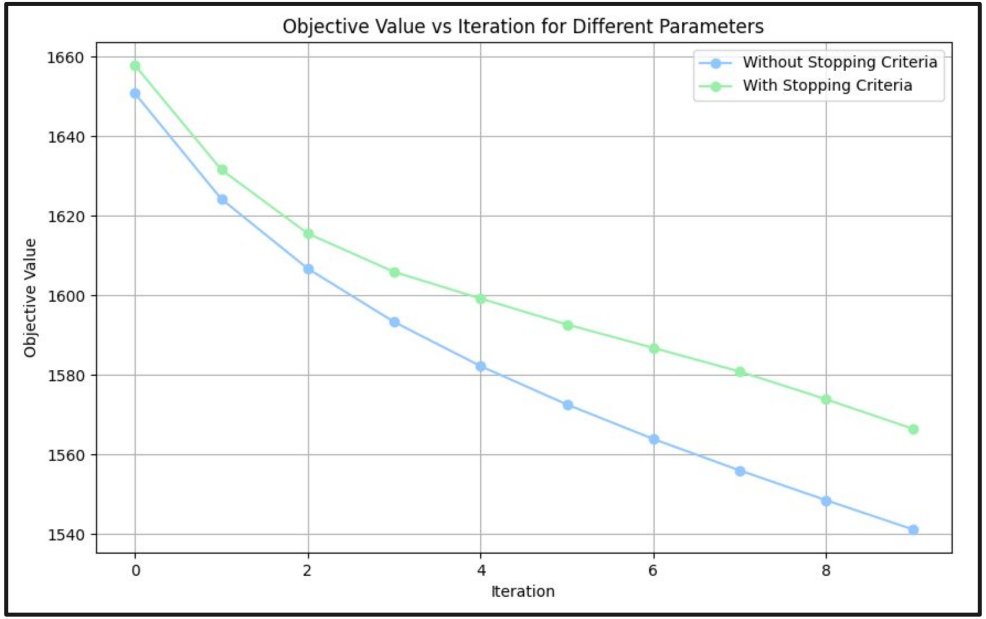


At the bottom 30%, $\lambda = 0$, and it increases gradually to 0.8 to the top



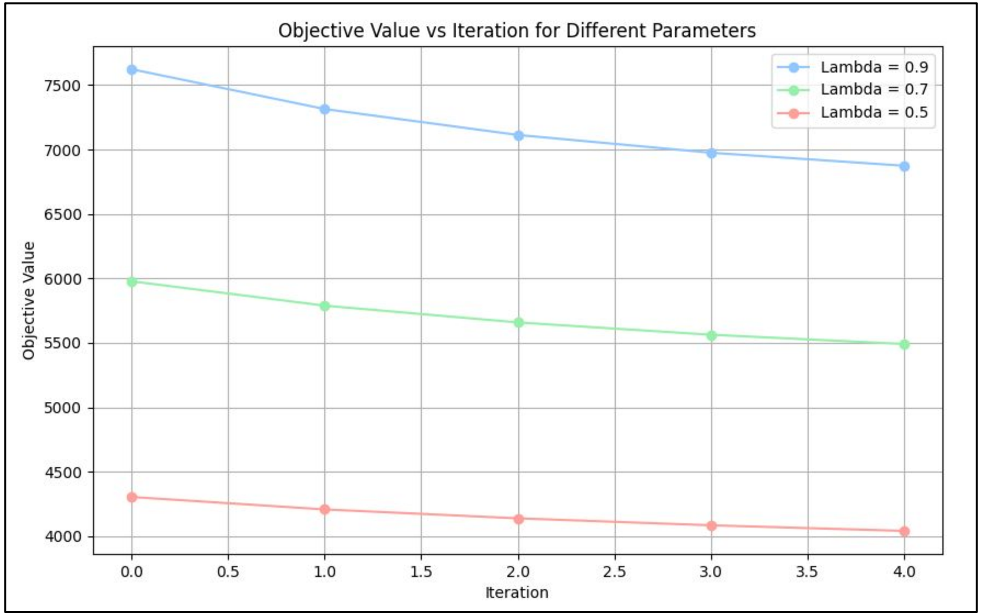
Lambda = 0.8 throughout

Experimenting with stopping criteria



Utilizing a stopping criteria does not significantly improve the objective value but achieves comparable results in lesser time.

Experimenting with various cubeness degree



A higher value of lambda means to a higher objective value, contributing to steeper contours.

Thank you!

Any questions?