DGP 第四次作业: Cubic Stylization

SA22001009 陈泽豪

May 24, 2024

1 作业介绍

Cubic Stylization 是一种用于图像处理的技术,旨在将图像转换为具有立体主义风格的艺术效果。立体主义(Cubism)是一种艺术流派,通过几何形状和多视角的表现手法来展现物体的不同侧面和结构。Cubic Stylization 通过计算机算法模仿这种艺术风格,将输入图像转换为具有类似效果的输出图像。Cubic Stylization 的主要作用是为图像添加独特的艺术风格,使其具有更加抽象和几何化的视觉效果。这种技术广泛应用于数字艺术、广告设计、以及各种创意视觉作品中,能够赋予普通图像新的艺术生命力和美感。具体如文章"Cubic Stylization"的 Teaser 内展示的:



Figure 1: Cubic Stylization 示意图

2 整体的算法框架

整个代码的实现主体实际上可以分为如下的几步。

2.1 建立起整体的能量函数

$$\min_{\tilde{V},\{R_i\}} \sum_{i \in V} \sum_{j \in \mathcal{N}(i)} \underbrace{\frac{w_{ij}}{2} \|R_i d_{ij} - \tilde{d}_{ij}\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \|R_i \hat{n}_i\|_1}_{\text{CUBENESS}} \tag{1}$$

方程在 ARAP 的基础上加入了 L_1 的 cubeness 约束项,通过控制 cubeness 项的系数就可以让最终的优化结果接近 cubic 的风格化结果。

2.2 Local Step

首先是固定住 \tilde{d}_{ij} 之后针对 R_i 的优化,因此这里我们不能简单的像 ARAP 那时候处理,需要用到 ADMM 的技术。假设我们针对每一个点进行分别的讨论,那么就可以建立如下的优化式子:

$$\min_{z, R_i \in SO(3)} \frac{W_i}{2} \|R_i D_i - \tilde{D}_i\|_F^2 + \lambda a_i \|z\|_1$$
subject to $z - R_i \hat{n}_i = 0$. (2)

虽然我觉得文中的这种写法存在问题,但是底下的优化出 R_i 的 ADMM 过程是正确的。

$$R_{i}^{k+1} \leftarrow \arg\min_{R_{i} \in SO(3)} \frac{W_{i}}{2} \|R_{i}D_{i} - \tilde{D}_{i}\|_{F}^{2} + \frac{\rho^{k}}{2} \|R_{i}\hat{n}_{i} - z^{k} + u^{k}\|_{2}^{2}$$

$$z^{k+1} \leftarrow \arg\min_{z} \lambda a_{i} \|z\|_{1} + \frac{\rho^{k}}{2} \|R_{i}^{k+1}\hat{n}_{i} - z + u^{k}\|_{2}^{2}$$

$$\tilde{u}^{k+1} \leftarrow u^{k} + R_{i}^{k+1}\hat{n}_{i} - z^{k+1}$$

$$\rho^{k+1}, u^{k+1} \leftarrow \operatorname{update}(\rho^{k})$$

$$(3)$$

更具体的来说,第一步的优化过程如下所示:

$$M_i = \begin{bmatrix} D_i & \hat{n}_i \end{bmatrix} \begin{bmatrix} W_i \\ \rho^k \end{bmatrix} \begin{bmatrix} \tilde{D}_i^\top \\ (z^k - u^k)^\top \end{bmatrix}, \tag{4}$$

$$M_i = \mathcal{U}_i \Sigma_i \mathcal{V}_i^{\top}, \tag{5}$$

$$R_i^{k+1} \leftarrow \mathcal{V}_i \mathcal{U}_i^{\top}, \tag{6}$$

如果得到的 R_i^{k+1} 行列式不大于 0,则给 U_i 矩阵的最后一列乘以 -1。第二步的优化如下所示:

$$z^{k+1} \leftarrow \mathcal{S}_{\lambda a_i/\rho^k} \left(R_i^{k+1} \hat{n}_i + u^k \right) \tag{7}$$

$$S_{\kappa}(x)_{j} = \left(1 - \frac{\kappa}{|x_{j}|}\right)_{+} x_{j} \tag{8}$$

第三步的优化就是值的输入。第四步的话 update 过程如下所示:

$$\rho^{k+1}, u^{k+1} := \begin{cases} \tau_{\text{incr}} \rho^k, \tilde{u}^{k+1} / \tau_{\text{incr}} & \text{if } || r^k ||_2 > \mu || s^k ||_2 \\ \rho^k / \tau_{\text{decr}}, \tau_{\text{decr}} \tilde{u}^{k+1} & \text{if } || s^k ||_2 > \mu || r^k ||_2 \\ \rho^k, \tilde{u}^{k+1} & \text{otherwise} \end{cases}$$
(9)

其中 r^k, s^k 表示 primal residual 以及 dual residual,定义如下:

$$r^k = \|z - R_i^k \hat{n}_i\|_2 \tag{10}$$

$$s^{k} = \| -\rho^{k}(z^{k} - z^{k-1})\|_{2} \tag{11}$$

因此不断的进行 ADMM, 直到两个 residual 小于某个值。或者自行设定 ADMM 的迭代次数。

2.3 Global Step

当固定住 R_i 之后,求新的顶点位置的思路实际上就与原先的 ARAP 一模一样。因此可以直接求偏导列公式求解。最终的算法就如文章内提到的:

Algorithm 1 Cube Stylization (λ)

Require: A triangle mesh V, F

Ensure: Deformed vertex positions \tilde{V}

1: $\tilde{V} \leftarrow V$

2: while not converge do

3: $R \leftarrow \text{local-step}(V, \tilde{V}, \lambda)$

4: $\tilde{V} \leftarrow \text{global-step}(R)$

5: end while

3 最终结果展示

实现采用 python 3.9.19, igl 2.2.1, numpy 1.26.4, scipy 1.13.0, open3d 0.18.0. 其中 open3d 用来显示网格结果。直接在 main.py 内点击运行即可输出输入的 mesh 以及 cubic stylization 结果,如下所示:

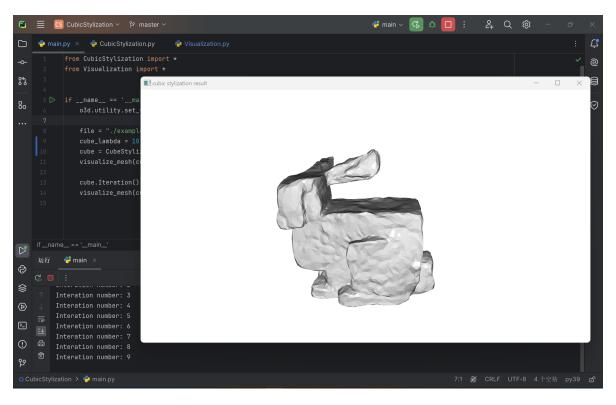


Figure 2: 运行示意图

3.1 示例

参数的设定都按照论文内的样子,我们只调整 cubeness 内的 λ 的值,得到如下所示的结果:

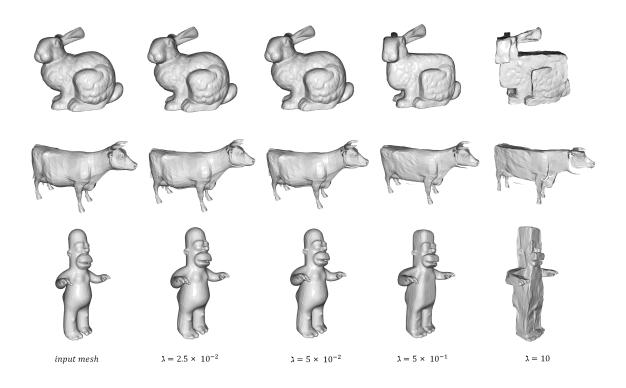


Figure 3: bunny 运行结果

4 总结

我们在这一次作业里基于 ARAP 能量函数,配合惩罚项实现了 cubic stylization,这个过程 里又使用了 ADMM 的技术,收获颇多。