
中文内容摘要

三维网格间插值是一个有广泛应用的计算机图形学问题。我们提出了一种基于边长的在两个具有相同连接关系的三维曲面间插值的方法。该方法主要是使用牛顿法对能量进行优化。并且我们通过将 **hessian** 矩阵近似为正交矩阵以提高优化速度。相比于已有的其它基于边长的插值方法，该方法具有更小的误差。

关键词：插值；形变

第一章 简介

三维网格间插值是一个很经典的图形学问题，可应用于很多方面，比如计算机动画等。而三维网格间插值可以基于许多标准，比如角度、变换矩阵、边长。我们这方法使用的标准即是边长。

第二章 算法

第一节 能量优化

设顶点 i 的坐标为 p_i , 则顶点 i 和顶点 j 连成的边 e_{ij} 的长度为 $\|p_i - p_j\|$. 我们的目标是优化能量使得最终的曲面的边长等于目标边长. 对于 e_{ij} , 定义能量

$$E_{ij} = \|\|p_i - p_j\|^2 - l_{ij}^2\|^2 \quad (1)$$

$$E = \sum_{e_{ij}} E_{ij} \quad (2)$$

其中 l_{ij} 是 e_{ij} 的目标边长.

我们使用牛顿法优化能量 E , 因此需要一阶梯度和二阶偏导. 设 $p_i = (p_{i1}, p_{i2}, p_{i3})$, 则通过计算可得,

$$\frac{\partial E_{ij}}{\partial p_{ik}} = 4(\|p_i - p_j\|^2 - l_{ij}^2)(p_{ik} - p_{jk}), \quad (3a)$$

$$\frac{\partial^2 E_{ij}}{\partial p_{ik}^2} = 8(p_{ik} - p_{jk})^2 + 4(\|p_i - p_j\|^2 - l_{ij}^2), \quad (3b)$$

$$\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{jk}} = -8(p_{ik} - p_{jk})^2 - 4(\|p_i - p_j\|^2 - l_{ij}^2), \quad (3c)$$

$$\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{il}} = 8(p_{ik} - p_{jk})(p_{il} - p_{jl}), \quad (3d)$$

$$\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{jl}} = 8(p_{ik} - p_{jk})(p_{jl} - p_{il}). \quad (3e)$$

其中 $k, l = 1, 2, 3$ 且 $k \neq l$. 观察可得,

$$\frac{\partial^2 E_{ij}}{\partial p_{ik}^2} = -\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{jk}}, \quad (4a)$$

$$\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{il}} = -\frac{\partial^2 E_{ij}}{\partial p_{ik} \partial p_{jl}}. \quad (4b)$$

设 H_{ij} 是 E_{ij} 的 hessian 矩阵, 则 H_{ij} 是一个 6×6 的对称矩阵. 由等式 (4a) 和 (4b) 可得 $H_{ij} = \begin{pmatrix} A_{ij} & -A_{ij} \\ -A_{ij} & A_{ij} \end{pmatrix}$, 其中 A 是一个 3×3 的对称矩阵, 即 $A_{ij} =$

$$4 \begin{pmatrix} d_{ij} + 2(p_{i1} - p_{j1})^2 & -2(p_{i1} - p_{j1})(p_{i2} - p_{j2}) & -2(p_{i1} - p_{j1})(p_{i3} - p_{j3}) \\ -2(p_{i1} - p_{j1})(p_{i2} - p_{j2}) & d_{ij} + 2(p_{i2} - p_{j2})^2 & -2(p_{i2} - p_{j2})(p_{i3} - p_{j3}) \\ -2(p_{i1} - p_{j1})(p_{i3} - p_{j3}) & -2(p_{i2} - p_{j2})(p_{i3} - p_{j3}) & d_{ij} + 2(p_{i3} - p_{j3})^2 \end{pmatrix},$$

其中 $d_{ij} = \|p_i - p_j\|^2 - l_{ij}^2$.

令 $H = \sum_{e_{ij}} H_{ij}$. 由于 H_{ij} 不一定是一个正定矩阵, 所以 H 也不一定是一个正定矩阵. 因此我们做适当的处理使得 H 是一个正定矩阵. 如果直接对 H 进行谱分解后正定化, 则由于 H 维数巨大, 计算量巨大, 因此不太合适. 考虑到 $H = \sum_{e_{ij}} H_{ij}$, 我们可以直接对 H_{ij} 进行正定化, 这样计算量小且能保证 H 是一个正定矩阵. 注意到 $H_{ij} = \begin{pmatrix} A_{ij} & -A_{ij} \\ -A_{ij} & A_{ij} \end{pmatrix}$, 我们可以利用这一特殊分布进一步减少计算. 我们先证明一个定理.

定理 2.1 设 $H = \begin{pmatrix} A & -A \\ -A & A \end{pmatrix}$, 则 H 是半正定矩阵的充分必要条件是 A 为半正定矩阵.

证明 充分性 若 A 是半正定矩阵, 则对任意向量 $x \in \mathbb{R}^n$ 有 $x^T A x \geq 0$. 对任意向量 $y \in \mathbb{R}^n$, 设 $y = (y_1, y_2)$, 其中 $y_1, y_2 \in \mathbb{R}^n$. 则有 $y^T H y = (y_1 - y_2)^T A (y_1 - y_2) \geq 0$. 因此 H 是半正定矩阵.

必要性 若 H 是半正定矩阵, 则对任意向量 $y \in \mathbb{R}^n$, 设 $y = (y_1, y_2)$, 其中 $y_1, y_2 \in \mathbb{R}^n$ 有 $y^T H y \geq 0$. 又因为 $y^T H y = (y_1 - y_2)^T A (y_1 - y_2)$, 因此 $(y_1 - y_2)^T A (y_1 - y_2) \geq 0$. 令 $y_2 = 0$, 则有 $y_1^T A y_1 \geq 0$. 又因为 y_1 是任意的, 因此 A 是半正定矩阵. ■

由上述定理可知我们只需令 A_{ij} 为半正定矩阵即可使得 H_{ij} 为半正定矩阵.

设

$$P_{ij} = \begin{pmatrix} (p_{i1} - p_{j1})^2 & -(p_{i1} - p_{j1})(p_{i2} - p_{j2}) & -(p_{i1} - p_{j1})(p_{i3} - p_{j3}) \\ -(p_{i1} - p_{j1})(p_{i2} - p_{j2}) & (p_{i2} - p_{j2})^2 & -(p_{i2} - p_{j2})(p_{i3} - p_{j3}) \\ -(p_{i1} - p_{j1})(p_{i3} - p_{j3}) & -(p_{i2} - p_{j2})(p_{i3} - p_{j3}) & (p_{i3} - p_{j3})^2 \end{pmatrix},$$

则易得 P_{ij} 的特征值分别为 0, 0 和 $(p_{i1} - p_{j1})^2 + (p_{i2} - p_{j2})^2 + (p_{i3} - p_{j3})^2$.

设 H_{ij} 的三个特征值分别为 $\lambda_1, \lambda_2, \lambda_3$. 注意到 $H_{ij} = 4(\|p_i - p_j\|^2 - l_{ij}^2)I + 8P_{ij}$, 因此有

$$\lambda_1 = 4(\|p_i - p_j\|^2 - l_{ij}^2),$$

$$\lambda_2 = 4(\|p_i - p_j\|^2 - l_{ij}^2),$$

$$\lambda_3 = 12(p_{i1} - p_{j1})^2 + 12(p_{i2} - p_{j2})^2 + 12(p_{i3} - p_{j3})^2 - 4l_{ij}^2.$$

设其对应的单位特征向量为 x_1, x_2, x_3 , 则由于 P_{ij} 是对称矩阵, 因此 $P_{ij} = \lambda_1 x_1 x_1^T + \lambda_2 x_2 x_2^T + \lambda_3 x_3 x_3^T$. 由于 P_{ij} 是一个半正定矩阵等价于 $\lambda_1, \lambda_2, \lambda_3 \geq 0$, 因此我们只需

让 $\widetilde{\lambda}_k = \max(\lambda_k, 0), k=1,2,3$, 然后令 $\widetilde{P}_{ij} = \widetilde{\lambda}_1 x_1 x_1^T + \widetilde{\lambda}_2 x_2 x_2^T + \widetilde{\lambda}_3 x_3 x_3^T$. 则此时 \widetilde{P}_{ij} 为半正定矩阵. 由于 $x_1 x_1^T + x_2 x_2^T + x_3 x_3^T = I$ 且 $x_1 = x_2$, 因此 $P_{ij} = \widetilde{\lambda}_1 (I - x_3 x_3^T) + \widetilde{\lambda}_3 x_3 x_3^T$. 而且计算可得

$$x_3 = \frac{(p_{i1} - p_{j1}, p_{i2} - p_{j2}, p_{i3} - p_{j3})}{\sqrt{(p_{i1} - p_{j1})^2 + (p_{i2} - p_{j2})^2 + (p_{i3} - p_{j3})^2}}.$$

将其代入上式即可得到 \widetilde{P}_{ij} . 接下来我们用 \widetilde{P}_{ij} 替代 P_{ij} 则有 H_{ij} 为半正定矩阵, 进而有 H 为半正定矩阵. 由于 H 为半正定矩阵, 并不能保证是一个正定矩阵, 尽管在绝大多数情况下是一个正定矩阵. 因此我们可以令 $\widetilde{H} = H + \epsilon I$, 其中 ϵ 为一很小的数, 比如 $\epsilon = 10^{-15}$.

将 H 近似为正定矩阵后我们就得到了下降方向 $d = -H^{-1} \Delta E$. 令 $E = f(x + \alpha d)$, 其中 x 是当前点的坐标, α 是步长. 由于 $f(x + \alpha d)$ 是一个关于 α 的四次函数, 因此我们可以对 $f(x + \alpha d)$ 求导, 并令 α 为满足 $f(x + \alpha d)' = 0$ 的最小正实数. 由于 $f(x + \alpha d)'$ 在 0 处的导数小于 0 且在 α 足够大时有 $f(x + \alpha d)' > 0$, 因此这样的 α 总是存在的.

第二节 插值

我们接下来要对源网格和目标网格的对应边长的平方进行线性插值以得到连续的变换. 即对时间 $t \in (0, 1)$, $l_t = \sqrt{l_1^2 + l_2^2}$, 其中 l_1, l_2 分别为源网格和目标网格的边长. 这样插值后得到的结果如图 1 所示. 可以看到中间的图显示出在某些时刻存在插值结果不够光滑, 自然的情况.

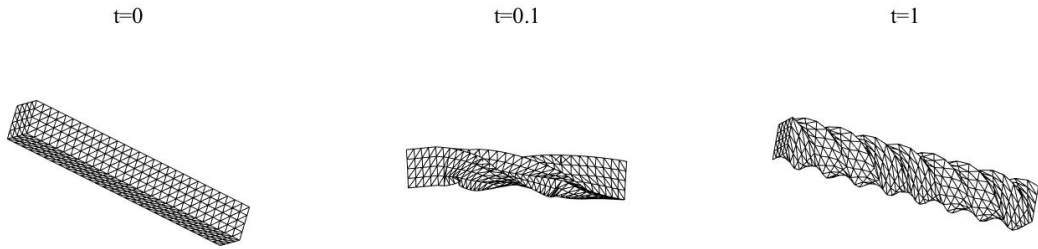


图 1: 上图是两个木条之间的插值结果. 其中左图和右图分别为初始网格和目标网格. 中图是 $t = 0.1$ 时的结果.

这是因为只有边长的连续性无法保证得到自然的插值结果, 还需要二面角

的连续性. 于是我们在之前的能量 E 的基础上加一项正则项, 即令

$$\tilde{E} = E + w\|P_{t_i} - P_i\|^2,$$

其中 t_i 为第 i 帧对应的时间 t , P_t 为 t 时刻所有点的坐标, P_i 为初始化网格, w 为权重. 我们取 $w = 10^{-3}l_{avg}^2$, 其中 l_{avg} 为网格的平均边长.

为了得到良好的插值结果, 我们需要一个合适的初始化网格. 于是我们比较了 FFMP[4], ARAP[2], GE[3], ABF[5], SQP[1] 共 5 种方式得到的网格. 首先, 我们先定义一个评价标准

$$L = \sqrt{\frac{1}{|E|} \sum_{e \in E} \left(\frac{l_{real}}{l_{target}} - 1 \right)^2},$$

其中 E 为所有边长的集合, l_{real} 是我们得到的网格的边长, l_{target} 是目标边长. 可以看到实际上 L 就是边长的相对误差的 l_2 范数.

		$L(10^{-3})$						
		airplane	armadillo	bar	bird	bust	camel	cat
FFMP	t=0.25	1.39	0.47	12.42	0.13	3.45	0.96	3.46
	t=0.5	2.50	0.86	4.59	0.18	3.44	1.45	3.85
	t=0.75	1.75	0.62	8.50	0.13	3.44	0.90	2.62
ARAP	t=0.25	2.57	3.35	11.17	1.20	1.73	2.98	6.29
	t=0.5	3.58	4.17	7.46	1.50	2.43	2.21	5.51
	t=0.75	2.36	3.88	5.03	0.96	1.76	1.61	5.55
GE	t=0.25	0.89	0.14	8.50	0.12	1.84	0.43	2.10
	t=0.5	0.65	0.16	9.62	0.16	2.21	0.48	2.09
	t=0.75	0.53	0.15	7.26	0.18	1.93	0.36	1.22
ABF	t=0.25	0.90	0.14	10.81	0.17	1.90	0.63	9.30
	t=0.5	0.59	0.18	13.80	0.23	2.70	0.74	7.50
	t=0.75	0.45	0.16	7.74	0.18	1.98	0.54	3.45
SQP	t=0.25	0.24	0.12	9.83	0.08	1.04	0.33	1.02
	t=0.5	0.29	0.15	8.74	0.10	1.24	0.38	0.81
	t=0.75	0.20	0.13	5.56	0.07	1.00	0.31	0.43

表 2.1 在不同初始化网格下得到的结果的 L 值

如表 2.1 所示，在这五种初始化网格中，SQP 得到的结果最好. 于是我们选择 SQP 得到的网格作为初始化网格.

第三章 实验结果

我们使用的系统为 Windows10,CPU 为 E5-2650v2, 内存容量为 32GB. 我们选取了 LSRDF[7], ELI[6] 和 MSGI[8] 用于对比.

图 2 展示了在 $t = 0.25, 0.5, 0.75$ 时在一些模型下的插值结果. 图 3 和图 4 则进一步展示了我们的方法与选取的三种方法的插值结果的比较. 可以看到我们的插值结果的边长误差均更小, 因此效果更好.

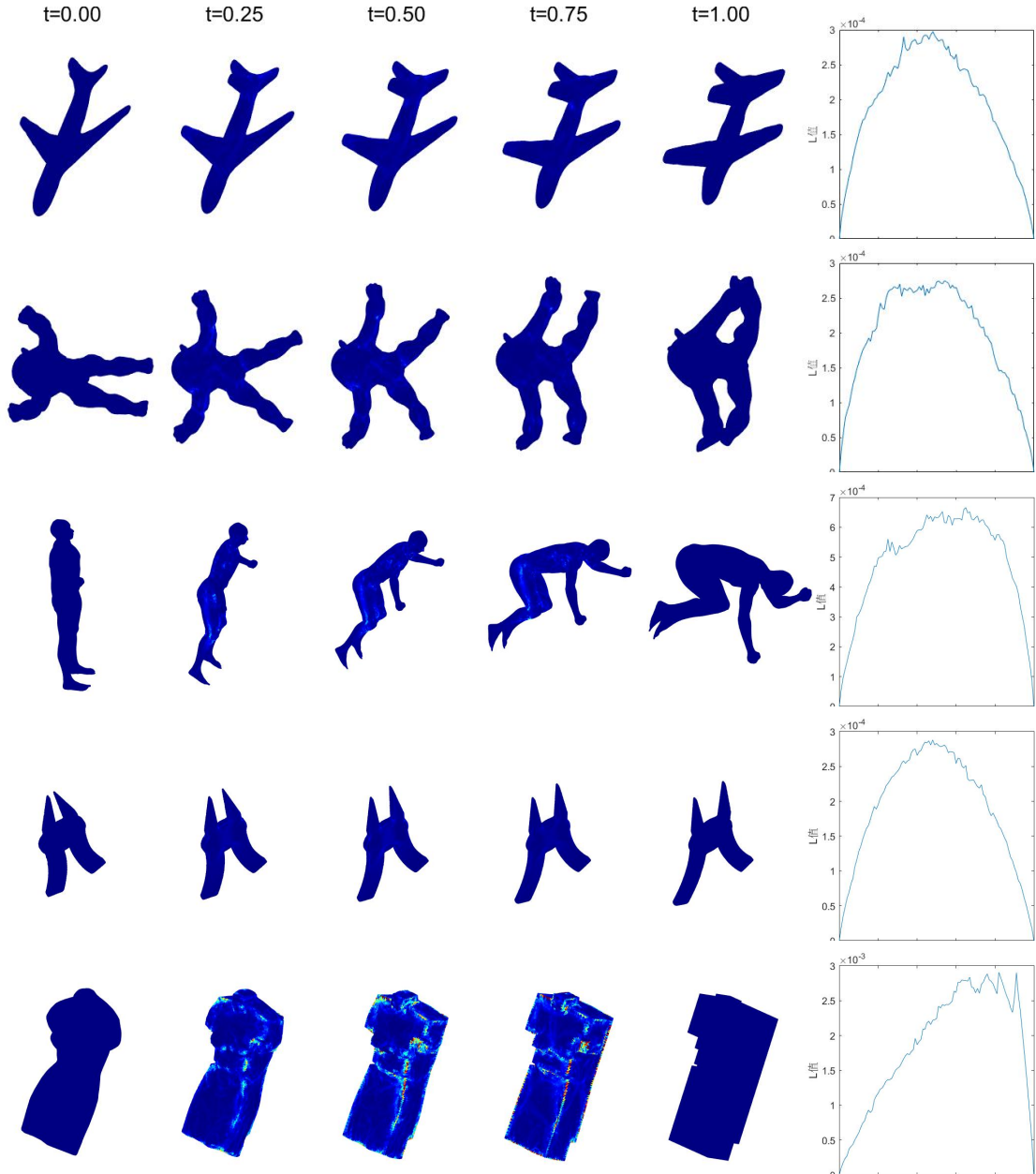


图 2: 飞机、狻猊、人、钳子和雕像这五个模型的插值结果。其中 $t=0.00$ 和 $t=1.00$ 分别代表源模型和模型, 其它时刻则为插值得到的结果. 最右边是不同时刻的 L 值误差.

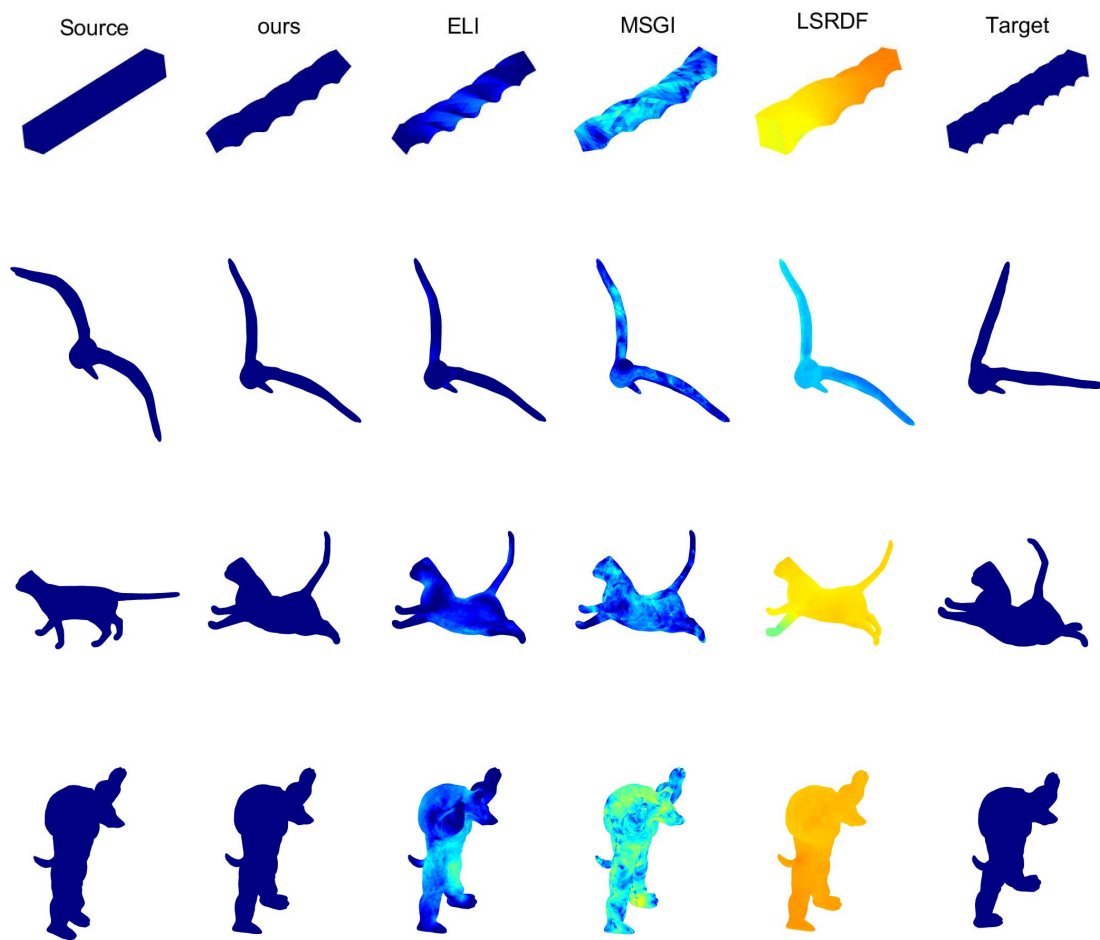


图 3: 比较四种方法在五个模型下的插值结果. 其中中间四列是四种方法在 $t = 0.5$ 时刻的结果. 颜色越偏橙红说明实际边长与目标边长的差越大, 越偏深蓝色则说明越小.

第四章 局限性

该方法的局限性主要有两个. 第一个是由于正则项包含初始化网格的信息, 因此结果的好坏部分依赖于初始化的好坏. 第二个是虽然总体上的边长的相对误差相较于其它方法有了较大减少, 但是最大的边长的相对误差并没有保证.

参考文献

- [1] AHARON, I., CHEN, R., ZORIN, D., AND WEBER, O. Bounded distortion tetrahedral metric interpolation. *ACM Transactions on Graphics* 38, 6 (2019), 182.1–182.17.
- [2] ALEXA, M. As-rigid-as-possible shape interpolation. In *SIGGRAPH2000 Conference Proceedings* (2000).
- [3] CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. A simple geometric model for elastic deformations. *Acm Transactions on Graphics* (2010).
- [4] KIRCHER, S., AND GARLAND, M. Free-form motion processing. *Acm Transactions on Graphics* 27, 2 (2008).
- [5] PAILLIER, G.-P., RAY, N., POULIN, P., LLA SHEFFER, A., AND LÉVY, B. Dihedral angle-based maps of tetrahedral meshes. *ACM Transactions on Graphics (SIGGRAPH 2015 Conf. Proc.)* 34, 4 (2015).
- [6] ROJAS, C., TSUI, A., HE, S., SIMONS, L., LI, S., AND AMENTA, N. Edge length interpolation.
- [7] WANG, Y., LIU, B., AND TONG, Y. Linear surface reconstruction from discrete fundamental forms on triangle meshes. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 2277–2287.
- [8] WINKLER, T., DRIESEBERG, J., ALEXA, M., AND HORMANN, K. Multi-scale geometry interpolation. *Computer Graphics Forum* 29, 2 (2010).