
中文内容摘要

形状插值问题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

第一节 引言

形状插值问题是一个很经典的图形学问题, 能将一个几何模型光滑的变换到目标模型. 其可应用于很多方面, 比如计算机动画 [15] 等. 而形状插值可以基于许多标准, 比如角度、变换矩阵、边长等 [14]. 对于主要基于边长的插值方法, 例如 [9, 1], 前者针对面网格, 后者则针对体网格. 它们最后得到的结果距离期望边长依然有较大距离, 不过后者得到的结果更好. 但是后者之所以在边长上难以更进一步, 是因为后者的处理对象是体网格, 若单纯插值边长, 其内部往往无法正好粘合. 因此, 若想使边长进一步优化, 内部的边会阻碍这一目标. 但是面网格则不会有这一困难. 并且考虑到前者方法 [9] 的插值结果的边长与期望边长的差距并不比后者方法 [1] 的插值结果小, 因此我们猜测关于面网格的方法在边长这一方面仍有很大的进步空间, 甚至可能得到与期望边长几乎一样的插值结果.

从自由度方面来看, 我们的猜测也是有道理的. 对于一个亏格为 0 的封闭三维面网格, 设其边, 顶点和面数分别为 e, v, f . 则由于每个面都是三角形, 因此有 $3f = 2e$. 另外由欧拉公式可知 $f + v - e = 2$. 因此在已知所有边长的情况下, 其自由度为 $3v - e = 3(e - f + 2) - e = 6$, 正好对应于刚性变换的 6 个自由度. 因此当边长确定时, 从自由度的角度来看, 存在满足条件的网格且网格各点的位置是确定的. 并且 Cauchy[3] 也证明了当凸的多面体网格的各个面是刚性时, 其整体是刚性的. Gluck[5] 之后则证明了大多数亏格为 0 的三角形网格是刚性的. 因此从这个角度来看, 我们更有理由相信当边长确定时, 存在至少一个网格满足边长条件, 且网格各点的位置是确定的. 基于以上理由, 我们尝试着在面网格上基于边长进行插值.

第二节 相关工作

形状插值问题研究已经有几十年历史了, 其相关工作非常多. Sedberg 等 [10] 早在 1993 年就研究了二维多边形插值问题. 如之前所说, 形状插值可以基于许多标准, 比如角度、仿射变换矩阵、边长. Alexa 等 [2] 提出了 ARAP 方法, 即是基于仿射变换矩阵进行形状插值. 该方法首先将仿射变换矩阵分解成旋转部分和放缩部分, 然后分别对旋转部分和放缩部分进行插值, 最后将这两部分进行混合, 得到了目标的仿射变换矩阵. Winkler 等 [12] 则利用了二面角和边长进行形状插值. 他们将网格分解成多个层次, 形成一棵树, 顶部根节点包含整个网格, 而底部节点则只有几个相邻的三角形面片. 然后将之从下到上层层地进行重建, 最终得

到插值结果.Wang 等 [11] 也是利用了二面角和边长进行形状插值. 不过, 他们并没有将网格分解成多个层次, 而是在每个面上建立了局部坐标系. 然后利用局部坐标系分别建立了二面角和边长对应的能量. 最终通过求解一个二次多项式的最小值获得最终的结果. 相比于刚才两种方法都利用了二面角, Rojas 等 [9] 则提出了一种只利用边长就得到插值结果的方法. 他们先利用边长求出满足条件的二面角, 再利用二面角得到每条边的旋转矩阵, 最后再求出网格顶点的位置. 不过由于该方法得到二面角后经过两步才得到最终的网格, 因此边长的误差会被放大.

第三节 算法

一、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 $i, j, k, l = 1, 2, 3$ 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 $A_{ij} =$

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

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. 事实上, 由舒尔补的性质 [13] 可知, 若 $H = \begin{pmatrix} A & -A \\ -A & A \end{pmatrix}$, 则 H 是半正定矩阵的充分必要条件是 A 为半正定矩阵.

由上述事实可知我们只需令 A_{ij} 为半正定矩阵即可使得 H_{ij} 为半正定矩阵. 设

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

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

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

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

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

由于 A_{ij} 是对称矩阵, 因此存在正交矩阵 Q 和对角阵 D 使得

$$A_{ij} = QDQ^T,$$

其中 $D = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$. 设为 $Q = (x_1, x_2, x_3)$, 则将上式展开可得

$$P_{ij} = \lambda_1 x_1 x_1^T + \lambda_2 x_2 x_2^T + \lambda_3 x_3 x_3^T.$$

由于 A_{ij} 是一个半正定矩阵等价于为 $\lambda_1, \lambda_2, \lambda_3 \geq 0$, 因此我们只需让 $\tilde{\lambda}_k = \max(\lambda_k, 0), k=1,2,3$, 然后令 $\tilde{A}_{ij} = \tilde{\lambda}_1 x_1 x_1^T + \tilde{\lambda}_2 x_2 x_2^T + \tilde{\lambda}_3 x_3 x_3^T$. 则此时 \tilde{A}_{ij} 为半正定矩阵. 由于 Q 是一个正交阵, 即 $QQ^T = I$. 因此有 $x_1 x_1^T + x_2 x_2^T + x_3 x_3^T = I$ 且 $x_1 = x_2$, 因此

$$P_{ij} = \tilde{\lambda}_1 (I - x_3 x_3^T) + \tilde{\lambda}_3 x_3 x_3^T.$$

接下来我们只需计算 x_3 . 经过计算可得

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

将其代入上式即可得到 \tilde{A}_{ij} . 接下来我们用 \tilde{A}_{ij} 替代 A_{ij} 则此时有 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. 为了确保 H 是一个正定矩阵, 在这里我们可以令 $\tilde{H} = H + \epsilon I$, 其中 ϵ 为一很小的数, 例如 $\epsilon = 10^{-15}$.

将 H 近似为正定矩阵后我们就得到了下降方向 $d = -H^{-1} \nabla E$. 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.

二、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.

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

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

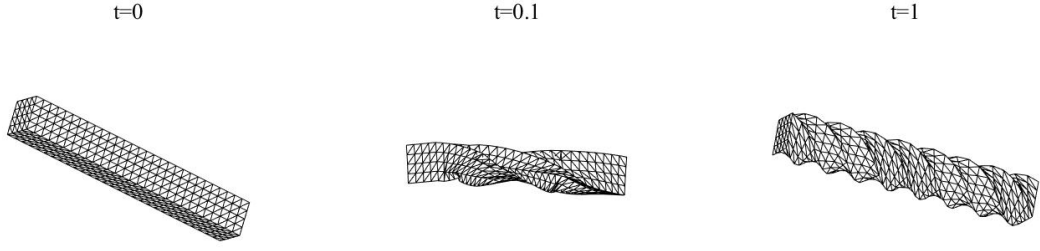


图 1: 上图是两个木条之间的插值结果. 其中左图和右图分别为初始网格和目标网格. 中图是 $t = 0.1$ 时的结果.

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. 考虑到我们的目标是优化边长, 因此我们自然觉得初始化的边长误差越小, 结果越好. 于是我们准备使用 SQP[1] 这个已经对边长进行较好优化的方法进行初始化. 为了验证我们的猜想, 我们还将其与 FFMP[6], ARAP[2], GE[4], ABF[7] 共 4 种方式得到的网格比较. 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.

如表 2.1 所示, 在这五种初始化网格中, SQP 的 L 值更小, 得到的结果的总体误差最小. 图 2 则是多个模型在使用不同初始化网格下得到的 $t=0.5$ 时刻的结果. 从图 2 则可以看到 SQP 得到的结果更蓝, 对应了其更小的 L 值. 并且我们可以看到 SQP 得到的结果更加合理. 因此, SQP 得到的结果最好, 于是我们选择其作为初始化网格.

第四节 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[11], ELI[8] 和 MSGI[12] 用于对比.

图 3 展示了在 $t = 0.25, 0.5, 0.75$ 时在一些模型下的插值结果. 图 4 和图 5 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. 从图 4 可以看到我们的方法在这几个模型下的边长误差均是最小的, 且整体上误差相近. 从图 5

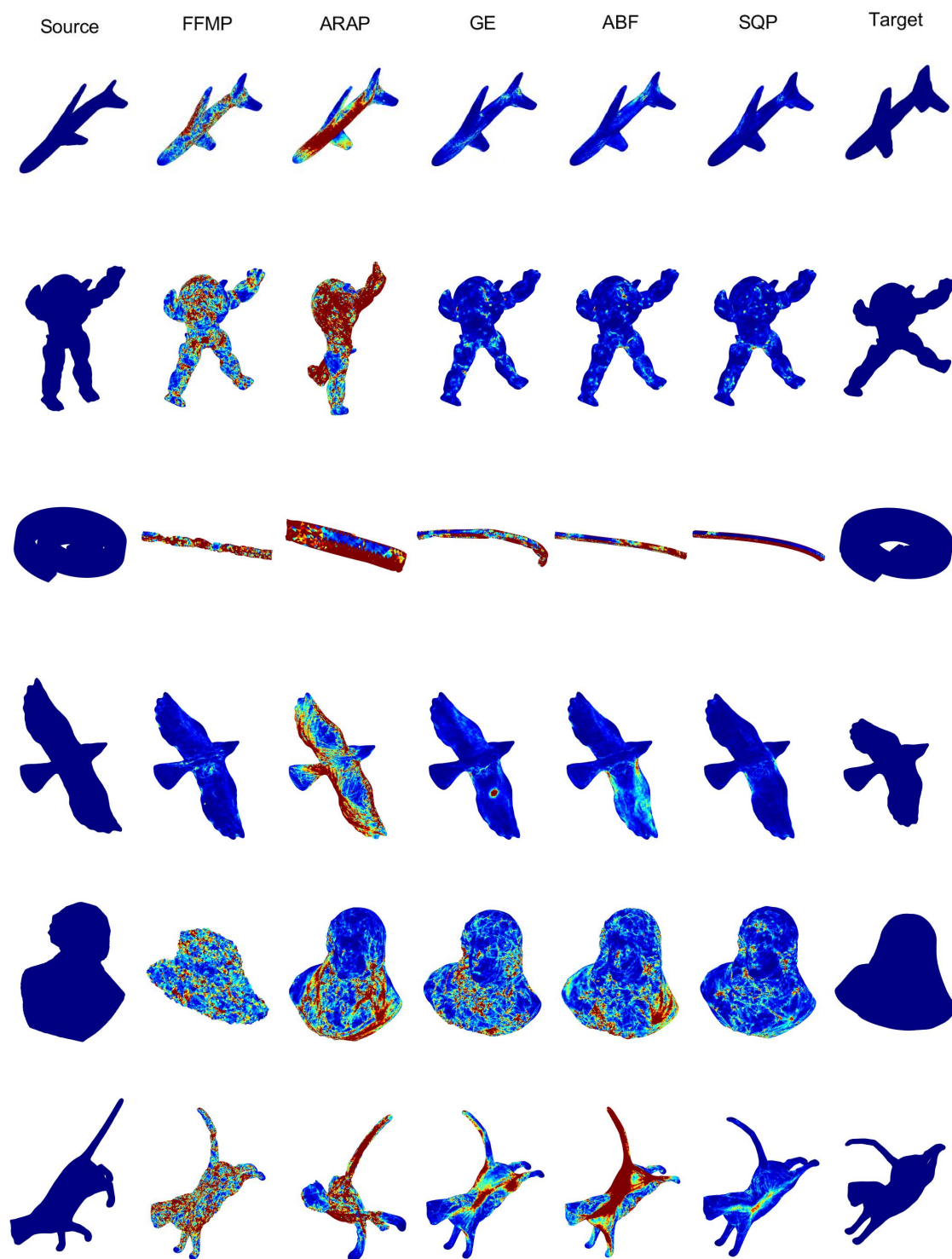


图 2: 在不同初始化网格下 $t=0.5$ 时得到的结果. 从上往下分别是对应于表 2.1 的 airplane, armadillo, bar, bird, bust, camel, cat 这 7 个模型. 颜色越偏橙红说明实际边长与目标边长的差越大, 越偏深蓝色则说明越小.

		$L(10^{-3})$						
		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 1 在不同初始化网格下得到的结果的 L 值

左列则可以看到我们的方法的 L 值更小, 即总体上来说我们的方法在各个形变的过程中边长误差均是最小的. 而从图 5 右列则可以看到除了在第一个模型, 即 **bar** 这个模型的某些时刻外, 我们的方法在各个形变的过程中最大的相对误差均是最小的. 因此无论是从图 4, 还是从图 5, 我们的方法均是最好的.

第五节 总结和讨论

从之前的实验可以看出, 我们的方法得到的结果, 其边长相比其它方法更加接近我们的期望, 误差更小. 在我们比较的方法中, 与我们最接近的是 ELI[8], 均只使用了边长信息. 但是由于我们的方法不需要像 ELI 一样进行多次转换得到最终结果, 因此能获得更小的误差. 不过该方法也有其局限性. 其局限性主要有两个. 第一个是由于正则项包含初始化网格的信息, 因此结果的好坏一定程度上依

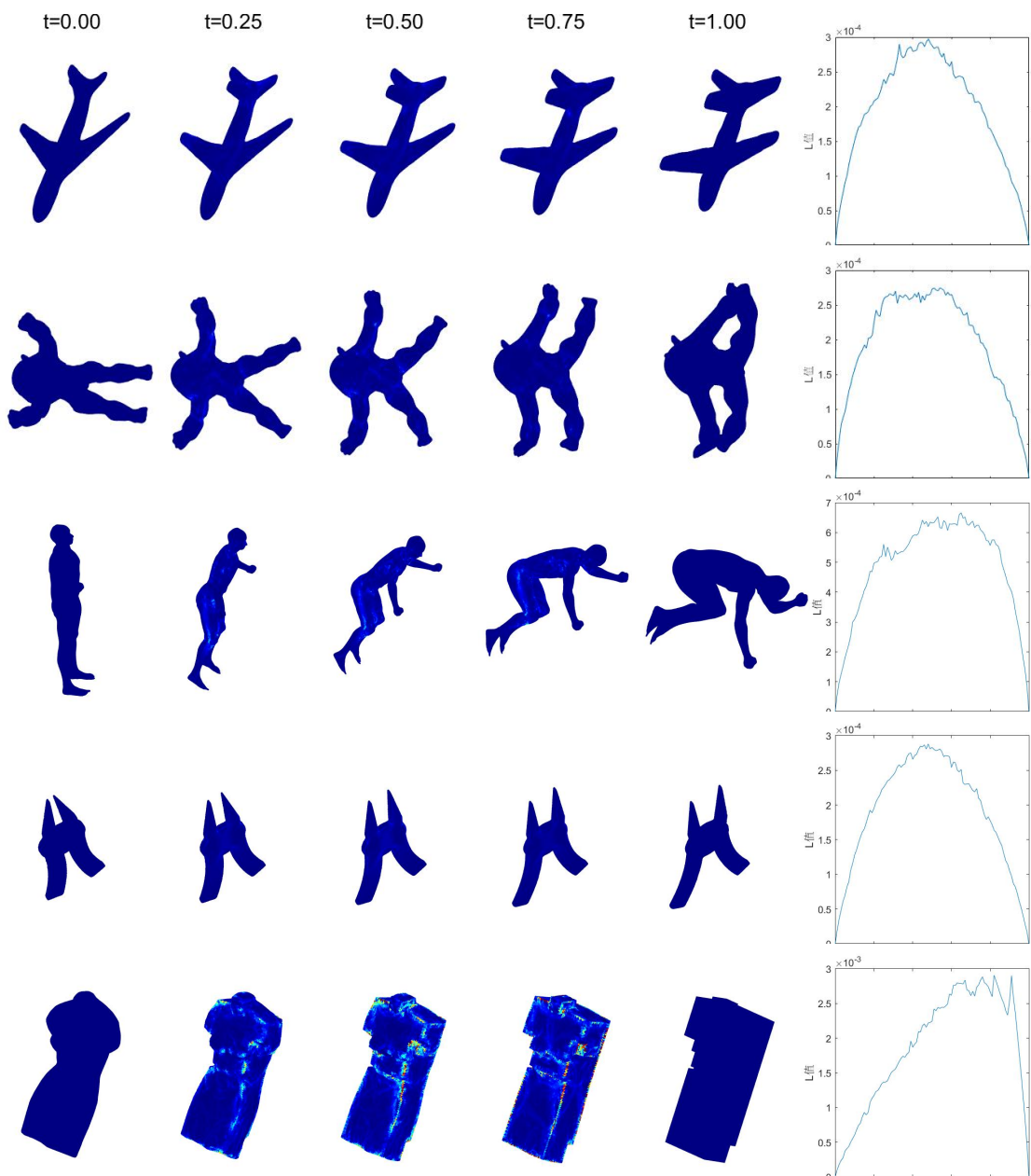


图 3: 飞机、狢狢、人、钳子和雕像这五个模型的插值结果. 其中 $t=0.00$ 和 $t=1.00$ 分别代表源模型和目标模型, 其它时刻则为插值得到的结果. 最右边是不同时刻的 L 值误差.

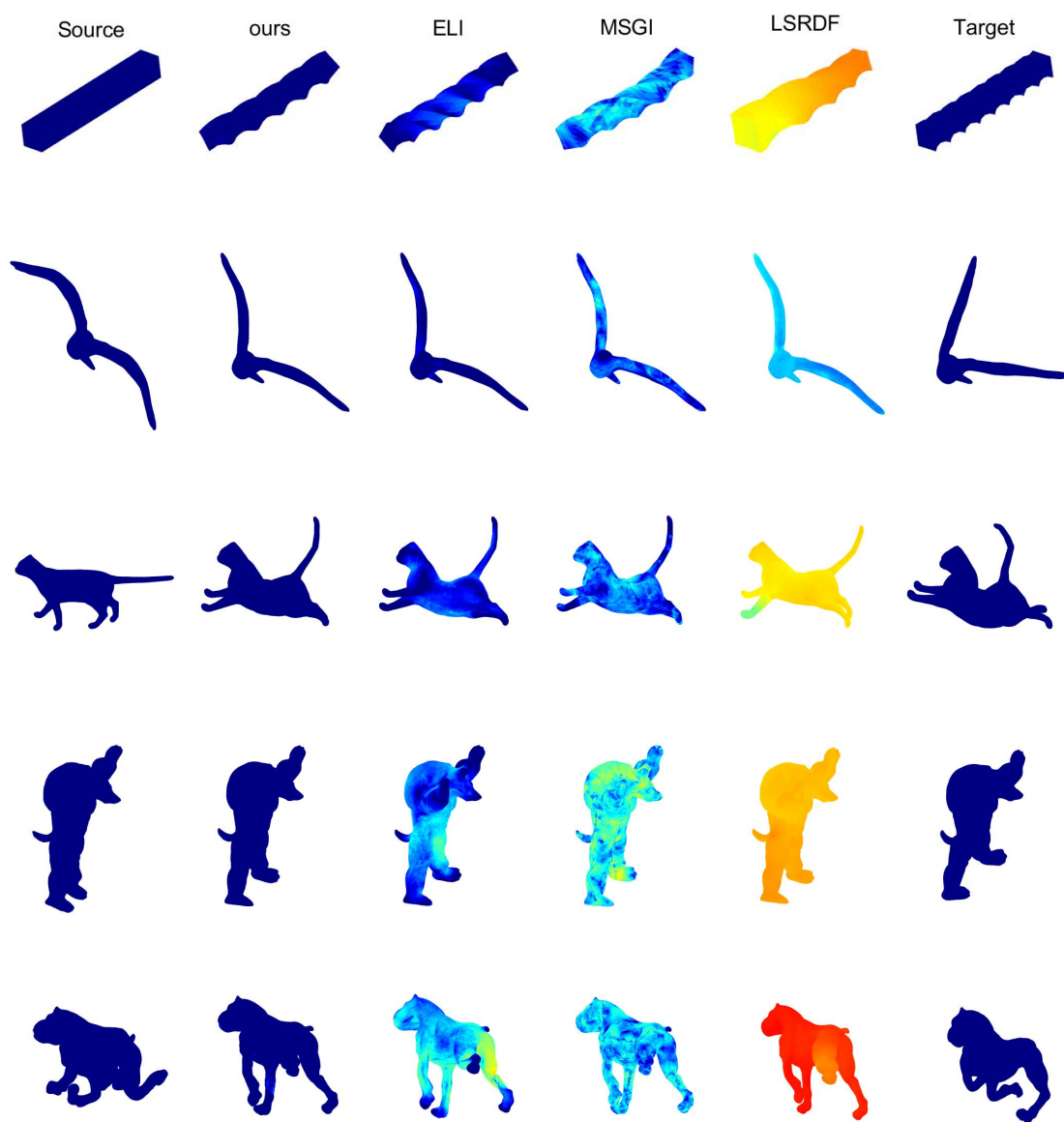


图 4: 比较四种方法在五个模型下的插值结果. 其中中间四列是四种方法在 $t = 0.5$ 时刻的结果. 颜色越偏橙红说明实际边长与目标边长的差越大, 越偏深蓝色则说明越小.

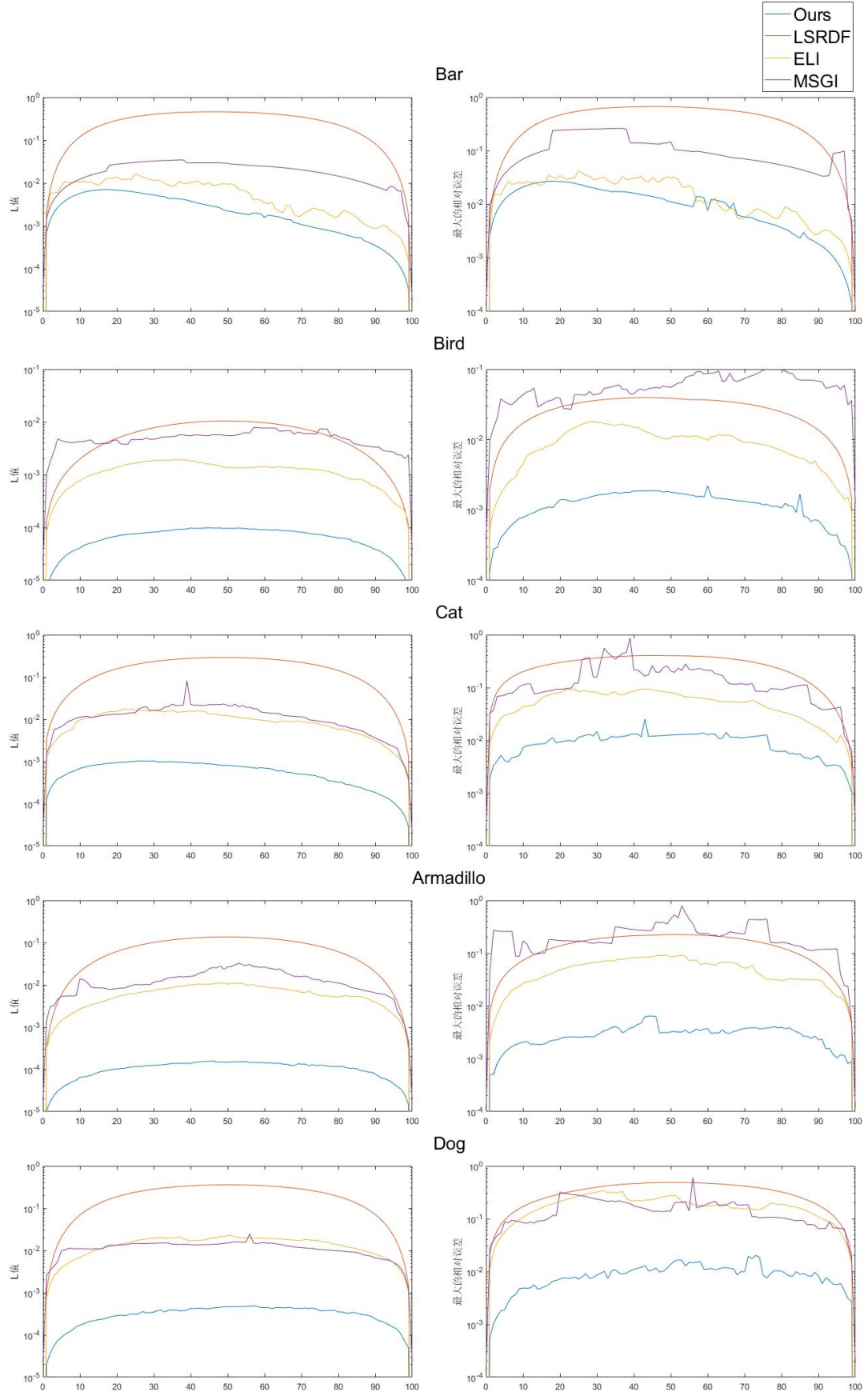


图 5: 比较四种方法在图三五个模型下的误差. 横轴代表帧数, 其中第 0 帧和第 100 帧分别代表源模型和目标模型. 左列是四种方法下的 L 值, 右列 2 则是最大的相对误差, 其中相对误差为 $|\frac{l_{real}-l_{target}}{l_{target}}|$.

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, 因此在少数情况下该指标不如其它方法. 而且我们最终结果的边长依然与我们期望的长度有一定差距. 因此, 我们接下来可以尝试寻找更好的初始化方法. 其次可以考虑是否有更好的优化方法并结合更好的初始化使得最终结果的边长等于我们期望的长度.

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] CAUCHY, A. L. Sur les polygones et polyedres.
- [4] CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. A simple geometric model for elastic deformations. *Acm Transactions on Graphics* (2010).
- [5] GLUCK, H. *Almost all simply connected closed surfaces are rigid*. Geometric Topology, 1975.
- [6] KIRCHER, S., AND GARLAND, M. Free-form motion processing. *Acm Transactions on Graphics* 27, 2 (2008).
- [7] PAILLÉ, 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).
- [8] ROJAS, C., TSUI, A., HE, S., SIMONS, L., LI, S., AND AMENTA, N. Edge length interpolation.
- [9] ROJAS, C., TSUI, A., HE, S., SIMONS, L., LI, S., AND AMENTA, N. Edge length interpolation.
- [10] SEDERBERG, T. W., GAO, P., WANG, G., AND HONG, M. 2-d shape blending: An intrinsic solution to the vertex path problem. In *Conference on Computer Graphics & Interactive Techniques* (1993).
- [11] 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.
- [12] WINKLER, T., DRIESEBERG, J., ALEXA, M., AND HORMANN, K. Multi-scale geometry interpolation. *Computer Graphics Forum* 29, 2 (2010).
- [13] ZHANG, F. *The Schur complement and its applications*, vol. 4. Springer Science & Business Media, 2006.
- [14] 张智邦, 李桂清, 韦国栋, 王宇攀, 路慧娜, AND 袁青. 形状插值算法综述. 计算

机辅助设计与图形学学报 27, 8 (2015), 12.

[15] 金小刚, AND 鲍虎军. 计算机动画技术综述. 软件学报 8, 4 (1997), 11.