Variable Positional Constraints for Laplacian Trajectory Editing

Thomas Nierhoff, Sandra Hirche, and Yoshihiko Nakamura

***Abstract*— Laplacian Trajectory editing has been proven to work well for motion imitation tasks where the shape of a given trajectory has to be maintained. Derived from computer graphics the discrete Laplace-Beltrami operator is used to encode local trajectory properties. A least-squares solution preserves the local trajectory properties during the trajectory retargeting process while also considering a weighted set of positional constraints. Focusing in this paper on various methods for imposing positional constraints it is investigated how the retargeting quality can be improved for movement imitation tasks in constrained environments.**

I. INTRODUCTION

Used and investigated for more than ten years in computer graphics, Laplacian mesh editing is nowadays a standard method for deforming surface meshes in a user-friendly and intuitive way [1], [2]. It has been shown in [3] that Laplacian editing can be also used to deform discrete trajectories in a similar fashion which makes it well suited for motion imitation problems a given trajectory has to be deformed. By encoding the local trajectory curvature using the discrete Laplace-Beltrami operator, a translational invariant trajectory description is obtained. During trajectory retargeting, C^1-continuity can be maintained while sufficing positional constraints in a least-squares sense (determined by a weighting factor ω).

So far only hard positional constraints are considered for trajectories, i.e. fixing only few points of the trajectory to defined positions with a large weighting factor ω. On the other hand, [4] proposed a different weighting scheme in which all points of a surface mesh a fixed to their original position with a rather small weighting factor ω.

The contribution of this paper is the adaption of the method proposed in [4] to Laplacian trajectory editing. It is shown how the resulting soft deformation behavior can be used for collision avoidance with a resulting behavior similar to the Elastic Strips framework [5]. In addition it is combined with Gaussian Mixture Regression (GMR). Here the spatial variance from multiple demonstrations determines a spatially varying weighting factor ω. This accelerates trajectory retargeting in a constrained environment as it makes an additional path search obsolete.

Thomas Nierhoff and Sandra Hirche are with the Institute of Automatic Control Engineering (LSR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany
{tn, hirche}@tum.de.
Thomas Nierhoff and Yoshihiko Nakamura are with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan
nakamura@ynl.t.u-tokyo.ac.jp.

II. APPROACH

A. Laplacian Trajectory Deformation

We assume a trajectory given the combination of a path in m dimensions ($m = 3$ for the remainder of this paper), described by the sampling points $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]^T \in \mathbb{R}^{n\times 3}$ and associated temporal information $t_i(\mathbf{p}_i)$. The trajectory is represented as an graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with each vertex v_i being associated with one sampling point $\mathbf{p}_i \to v_i$,. The neighbor set $\mathcal{N}_i$ of the vertex v_i is the set of all adjacent vertices v_j. Accordingly the edge set is defined as $\mathcal{E} = \{e_{ij}\}, i, j \in \{0,..,n\}$ with

$$e_{ij} = \begin{cases} w_{ij} & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

and $w_{ij} = 1$ as uniform weights.

Instead of working in absolute Cartesian coordinates, the discrete Laplace-Beltrami operator $\boldsymbol{\delta}$ is used for specifying the local trajectory deformation [6]. For vertex v_i it is defined as

$$\boldsymbol{\delta}_i = \sum_{j\in\mathcal{N}_i} \frac{w_{ij}}{\sum\limits_{j\in\mathcal{N}_i} w_{ij}} (\mathbf{p}_i - \mathbf{p}_j), \tag{2}$$

The topology of the graph is defined by the Laplacian matrix matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ with

$$\mathbf{L_{ij}} = \begin{cases} 1 & \text{if } i = j, \\ -\dfrac{w_{ij}}{\sum\limits_{j\in\mathcal{N}_i} w_{ij}} & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

When concatenating all $\boldsymbol{\delta}_i$-values (called *Laplacian coordinates*) into a single vector as $\boldsymbol{\Delta} = [\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \ldots, \boldsymbol{\delta}_n]^T$, it can be rewritten as

$$\mathbf{LP} = \boldsymbol{\Delta}. \tag{4}$$

By imposing a set of p additional positional constraints in the form $\mathbf{p}_i = \mathbf{c}_i$, the system (4) can be solved for $\mathbf{P}_n$ using least squares

$$\begin{pmatrix} \mathbf{L} \\ \bar{\mathbf{P}} \end{pmatrix} \mathbf{P}_n = \begin{pmatrix} \boldsymbol{\Delta} \\ \mathbf{C} \end{pmatrix}, \tag{5}$$

with the definition of $\bar{\mathbf{P}} \in \mathbb{R}^{n\times n}$ and $\mathbf{C} \in \mathbb{R}^{n\times 3}$ as follows

$$\bar{P}_{ij} = \begin{cases} \omega_i & \text{if } i = j \text{ and } \mathbf{p}_i = \mathbf{c}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$C_{i:} = \omega_i \left[p_{ix}, p_{iy}, p_{iz}\right], \tag{7}$$

and the scalar weighting factor ω_i.

B. Obstacle Avoidance

The method can be used to avoid dynamic obstacles while maintaining the original trajectory shape in a least-squares sense. The presented approach consists of three parts

- A repulsive force for obstacle avoidance
- An attracting force pulling the trajectory back to its original position
- The Laplacian framework maintaining the local trajectory shape

Assume we are given a obstacle Ω with random shape. Let $\mathbf{d}_i$ be the shortest distance between a sampling point $\mathbf{p}_i$ of the trajectory and the obstacle Ω. Then the obstacle exerts a repulsive force on each point $\mathbf{p}_i$ according to

$$[p_{ix}, p_{iy}, p_{iz}] = \mathbf{p}_i + \alpha \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} \frac{1}{\|\mathbf{d}_i\|^2}, \quad (8)$$

$$\omega_i = \beta, \quad (9)$$

with constants α, β. Similar to potential fields the strength of the repulsive force decreases with a factor $\frac{1}{\|\mathbf{d}_i\|^2}$. The attractive force pulling the trajectory back is described using each sampling point's original position $\mathbf{p}_i^o$ before deformation

$$[p_{ix}, p_{iy}, p_{iz}] = \mathbf{p}_i^o, \quad (10)$$

$$\omega_i = \gamma, \quad (11)$$

with constant γ. By concatenating the matrices in Eq. 9,13 and solving Eq. 5 for $\bar{\mathbf{P}}$, the desired behavior can be achieved. The method shares some properties with the Elastic Strips framework. Still we think that our method is easier applicable to arbitrary-shaped trajectories.

C. Variance-Based Trajectory Deformation

Given a motion imitation task, the spatially changing variance between multiple demonstration trajectories can be combined with Laplacian Trajectory Editing for quick adaption to changed environmental constraints. A covariance matrix $\boldsymbol{\Sigma_i}$ with corresponding volume $o_i = \sqrt{\det(\boldsymbol{\Sigma_i})}$ is given for each sampling point $\mathbf{p}_i$. Then the reciprocal relation

$$\omega_i = \frac{\delta}{o_i}, \quad (12)$$

with constant factor δ ensures that trajectory segments with small variance keep their shape during the retargeting process while trajectory segments with large variance are deformed more.

III. Experiments

Two experiments are conducted, one for obstacle avoidance and another one for variance-based trajectory deformation.

In the first experiment, see Fig. 1, an initially sinusoidal trajectory is deformed as more and more round obstacles are added. The obstacle are avoided while the original trajectory shape is maintained. To prevent drift, the first and last sampling points are fixed using a large ω-value (1e9). The constant factors have been set to $\alpha = 0.5$, $\beta = 0.05$, $\gamma = 0.05$.

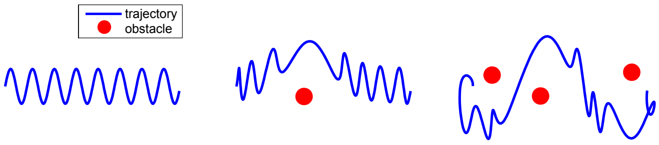

Fig. 1. Obstacle avoidance of a sinusoidal trajectory in the presence of multiple round obstacles

In the second experiment two rectangular areas are connected through a narrow tunnel, see Fig. 2. The RRT* algorithm is used to find sample trajectories between random positions in the right and left rectangular area. The sample trajectories are then normalized in the spatial domain using Dynamic Time Warping. A regressed trajectory with corresponding spatial covariance is obtained through Gaussian Mixture Regression. If the tunnel position changes from $\mathbf{p}_t^o$ to $\mathbf{p}_t$, the displacement is propagated to all trajectory sampling points as

$$[p_{ix}, p_{iy}, p_{iz}] = \mathbf{p}_i + \mathbf{p}^t - \mathbf{p}_t^o, \quad (13)$$

with ω_i according to Eq. 12. Similar to the previous example, the first and last sampling point of the trajectory are set to fixed positions using a large ω-value (1e9). One can see how a collision-free path is be obtained for multiple tunnel positions without the need to rerun the RRT* algorithm, resulting in a considerable time saving.

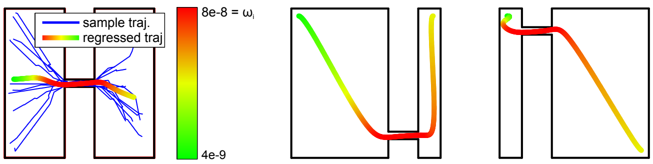

Fig. 2. Variance-based deformation. Left side: Sample trajectories and regressed trajectory with spatially varying weighting factor ω_i. Mid / right side: Adaption to changed environment

IV. Conclusion

In this paper we propose two different adaptive weighting schemes for Laplacian trajectory editing. Various, frequently occuring problems at trajectory retargeting are solved. Being easy to implement and robust during usage it is a viable alternative/extension to existing approaches.

References

[1] M. Alexa, "Differential coordinates for local mesh morphing and deformation," *The Visual Computer*, vol. 19, no. 2, pp. 105–114, 2003.

[2] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *EUROGRAPHICS*, 2007, pp. 109–116.

[3] T. Nierhoff and S. Hirche, "Fast trajectory replanning using laplacian mesh optimization," in *IEEE ICARCV*, 2012.

[4] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *ACM SIGGRAPH*, 2006, pp. 381–389.

[5] O. Brock and O. Khatib, "Elastic Strips: A Framework for Motion Generation in Human Environments," *Int. J. Rob. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.

[6] Y. Lipman, O. Sorkine, M. Alexa, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel, "Laplacian framework for interactive mesh editing," *IJSM*, vol. 11, no. 1, pp. 43–62, 2005.