Robust 3-D Field Line Query Based on Data Fusion of Multiple Leap Motions

Shunlong Ye, Guang Yang, Ziyu Yao, Xueyi Chen, Ting Jin, Genlin Ji, and Richen Liu *
School of Computer Science and Technology, Nanjing Normal University

Abstract

3-D field line data analysis is widely applied in multiple fields, such as steamline or pathline representing air movement in meteorological field, or Diffusion Tensor Imaging (DTI) showing water movement in medical field. Domain experts often need to query certain curve features in the densely sampled 3-D field lines to filter out unrelated lines. Extraction of such features often depends on experts' experience, which is of great significance for the analysis of data sets and the speculation of the hidden phenomena behind them.

Appropriate interaction design and realization would help users input and extract features. Currently, feature-based interactive field line visualization mostly uses traditional mouse drawing for 2-D curve input. However, 2-D curve often fails to represent the spatial features of the 3-D field lines. As a result, 2-D curve feature may result in low query accuracy especially for large-scale high-dimensional vector fields, which increase users' cognitive burden. In this paper, we propose a 3-D field line visualization and query system based on multiple Leap Motions. Users are allowed to input 3-D data with finger movement in a natural way, as well as basic visual interactions such as rotation, scaling, and translation. Moreover, the feature extraction algorithm is designed to query the field line in the flow field data set.

To demonstrate the effectiveness and accuracy of our method for flow field data exploration, we conduct three evaluation tests, the first is pathline data set analysis, the second is DTI data set analysis, and one for gesture trajectory detected by dual leap motions comparing with single leap motion. Tests show that 3-D curve input by multiple leap motions greatly improves input accuracy and feature extraction.

1 Introduction

3-D field line data analysis has a broad application prospect in multiple fields like meteorology and medicine. There are different kinds of field line data, such as meteorological pathline, steamline and medical Diffusion Tensor Imaging (DTI). These densely sampled 3-D field lines has different features and patterns, which contains scientifically meaningful information. As a result, it is critical to extract its features to describe and characterize them to enable users to make effective decision [1]. This process is called feature-based field line visualization. Since such hidden feature extraction methods often depends on experts' experience instead of specific mathematical or physical definitions, so good visual interaction to present features is required to utilize experts' knowledge to maximize [5].

The traditional interactive visualization is mainly achieved by screen and mouse. These traditional mouse-screen operations are confined to a 2-D plane, which has many limitations for 3-D data sets. However, in the traditional interactive field line visualization, complex high-dimensional field line data is often projected into 2-D space. The 2-D interactive method may work well for 2-D features, but fails to represent the spatial features of 3-D field lines, and sometimes result in unwanted or biased query. However, with the help of some 3-D intraction devices, exploration of 3-D dataset may be intuitive, immersive and effective.

In this paper, we propose a 3-D field line visualization and query system based on multiple Leap Motions. Leap Motion is a lightweight, cheap but powerful hand gesture detection device. It can detect hand bones' 3-D spatial coordinates fastly and accurately. We implement basic interaction and visualization for 3-D field line of pathline, steamline and DTI, as well as 2-D curve input via mouse and 3-D curve input via Leap Motions. Users can input curve features by just moving index finger, which is intuitive and effective. We also implement a feature-extraction algorithm to query field lines in real-time. Users can adjust several parameters of the algorithm to fine tune the query result.

Although Leap Motion has excellent visual image processing algorithm to provide fast and powerful hand gesture recognition, it may still has blind area when certain gestures block its sight. Especially when fingers overlap each other from a single Leap Motion's point of view. Aiming at this, we allow users to place two Leap Motions with different angles of view, and implement a data fusion algorithm that receives two Leap Motions' detection result to provide a more accurate gesture recognition.

2 Our Method

In this paper, we propose a field line visualization system that allows experts to input 3-D curves for feature query via Leap Motion. Experts can input complex curves by just moving their fingers in the detected area. More over, our system can connect two Leap Motions detecting hand gestures from different angles, which better avoids finger occlusion and can calculate fingers' positions with more accuracy. Fig. 1 shows the pipeline of the proposed approach.

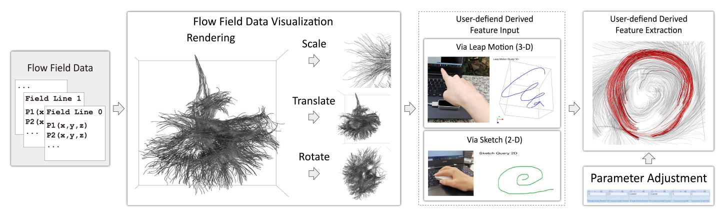

Figure 1: Firstly, the user loads the raw data and connect either one or two Leap Motions. Secondly, we visualize the field line data in the main view. Users can perform basic interactions like rotation, translation and scaling. Thirdly, users can input query curves via mouse or Leap Motion. The queried results are highlighted in the main view. Finally, users can adjust query parameters for more accurate results.

2.1 Field Line Visualization

During preprocessing, we sample the raw data, calculate the central point and the scale to make sure rendered image suits the main view. Then based on the method proposed by Chen et al. [3], we implement field line visualization and interaction with OpenGL. We orderly connect the 3-D sampling points on each field line to form curves, and draw them to the center of the box. The color and shadow effect will be rendered by OpenGL to convey spatial information. Users can also perform basic interaction such as rotation, scaling and translation by mouse dragging, rolling mouse roller and dragging with mouse roller pressed. The corresponding transformation matrixes are applied to the view.

2.2 Curve Input and Query

After basic interaction, users are allowed to input a curve to query field lines of similar shape. To input a 2-D curve, users can simply draw a curve with mouse in the mouse input view. To input a 3-D curve, users need to firstly spread fingers with palm down, and then draw the curve with index finger. After drawing is completed, users also need to spread fingers with palm down. We capture the index finger's movement, and visualize the curve on the input view. Then we use feature extraction algorithm to query the field line data set, and highlight the query result. Users can change certain parameters of the algorithm to adjust query results.

Specifically, to implement feature extraction algorithm, we adopt LCSS algorithm for two feature sequences: vector intersection angle feature and curvature feature. Firstly, we calculate the vector intersection angle for each

*The first two authors (undergraduates) contributed equally. Corresponding author: Richen Liu, e-mail: richen@pku.edu.cn. This work was supported by National Nature Science Foundation of China (61702271, 41971343)

of the sample point. In terms of curvature calculation, we adopt a discrete geometry algorithm based on An et al [2] to estimate the curvature for our discrete curves. To better calculate similarity, we classify intersection angle and curvature values into different data blocks by granularity. For example, when angle granularity is 10°, 82°and 87°are assigned with the same No.8 data block ID. As a result, we get a mapping between each sample field line and their corresponding two feature sequences. Then we apply LCSS algorithm [4] for the above two feature sequences to calculate similarity of the input curve and the flow curve. The LCSS algorithm calculates the longest common subsequence of a set of sequences. This method reduces storage use and makes query result irrelevant to curve length. We let P and Q be the feature sequence of two curves with length M and N, while *Id(P,i)* represents the feature's assigned block ID. The whole recursive LCSS algorithm is shown in Equation 1.

$$L(i,j) = \begin{cases} 0 & i = 0 \; or \; j = 0 \\ 1 + L(i-1, j-1) & i,j > 0 \; and \; Id(P,i) = Id(Q,J) \\ max(L(i-1,j), L(i,j-1)) & i,j > 0 \; and \; Id(P,i) != Id(Q,J) \end{cases} \quad (1)$$

We filter all the field lines with thresholds for intersection angle sequence and curvature sequence. Users can not only adjust thresholds for filter range, but also set different granularity for more precise or vague clasification of the feature. The query results are highlighted in real-time when editing these parameters.

2.3 Interaction via Two Leap Motions

Leap Motion can detect the location of every bone of users hand, but may have less accuracy for certain gestures, especially when the palm is nearly perpendicular to the Leap Motion, as shown in Fig. 2(b)(c). As a result, a second Leap Motion are suggested for more accurate detection. The assistant Leap Motion should be placed on the side of users' hand to obtain finger location that may not be visible for the main Leap Motion, as shown in Fig. 2(d)(e). We calculate the rotation matrix with the intersection angle of two Leap Motions, and then receives the finger bone coordinates from the assistant Leap Motion. We then transform these coordinates to the main Leap Motion's coordinate system with the rotation matrix, and calculate the average coordinates from two devices as the final output.

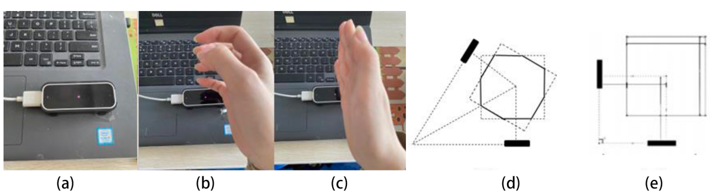

Figure 2: (a) is a single Leap Motion placed parallel to a desk. (b) and (c) show two gestures that Leap Motion may have trouble detecting. (d) and (e) show the positions of two Leap Motions and their common detecting area.

3 Results, Discussion and Conclusion

The first and second experiments mainly discuss the effects of 3-D gesture query based on somatosensory device Leap Motion for user-defined feature extraction in field line data set.

Fig. 3 shows the result of feature extraction of the ring structures in pathline data and linear structures in DTI data. We compares 3-D input method with traditional mouse drawing.

The third experiment tested the improvment of exploring medical DTI data with dual Leap Motion compared with single Leap Motion. We showed the detected tracks of different hand gestures. Results showed that using multiple Leap Motion could make hand detection more accurate as well as improve gestures recognition, which helps to boost efficiently and achieve an immersive interactive experience.

Fig. 4 shows the gesture trajectory in single Leap Motion and dual Leap Motion cases.

Limited by the performance of somatosensory devices, jitter may occur in the process of collecting user input data. This situation can be alleviated by placing multiple devices from different angles for data fusion, thus providing better user input experience. In the future, we want to apply this

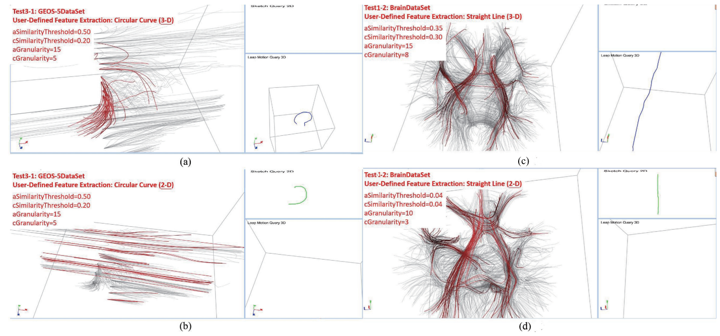

Figure 3: Both for feature extraction of ring structures in pathline data and linear structures in DTI data, a 3-D input method (a,c) results are significantly better than traditional mouse drawing (b,d). The ring structure feature in the pathline data are clearly extracted in (a), while the ring structure feature in the pathline data are not well reflected in (b). The linear trajectory is mainly in a symmetrical form clearly distributed on the left and right sides in (c), while in (d), the remaining ring structure feature (mainly located above and below the data set) are difficult to remove by parameter adjustment, resulting in visual confusion.

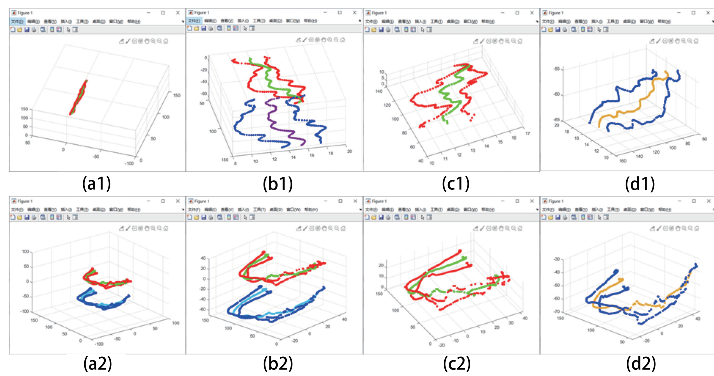

Figure 4: (a) The two tracks with the same color are the tracks of palm and the tracks of the center of palm and five fingertips detected by the primary or secondary Leap Motion, The track with different color is revised with data fusion. (b) Enlarged version of column 1. (c) Two red tracks are the track of the palm detected by the primary and secondary Leap Motion. The green one is revised with data fusion. (d) Two blue tracks are the tracks of the center of five fingertips and the palm detected by the two Leap Motion respectively. Yellow track is revised with data fusion.

3-D interaction method for seismic data [6]. Although two Leap Motions are used in our method, the Leap Motions' relatively small detecting range is still a limitation. When there is a large number of gestures or personnel involved, Leap Motions will not be able to correctly identify. In the future, we may expand the detection area by placing 3,4,5 or more Leap Motion in different positions, so as to form a larger detection area.

References

[1] Xueyi Chen a, Liming Shen a, Ziqi Sha a, Richen Liu a, Siming Chen b c, Genlin Ji a, and Chao Tan a. A survey of multi-space techniques in spatio-temporal simulation data visualization. *Visual Informatics*, 3(3):129–139, 2019.

[2] Y. An, L. Wang, R. Ma, and J. Wang. Geometric properties estimation from line point clouds using gaussian-weighted discrete derivatives. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2020.

[3] W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1433–1440, 2009.

[4] R. Liu, H. Guo, J. Zhang, and X. Yuan. Comparative visualization of vector field ensembles based on longest common subsequence. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 96–103, 2016.

[5] Richen Liu, Hanqi Guo, and Xiaoru Yuan. User-defined feature comparison for vector field ensembles. *Journal of Visualization*, 2017.

[6] Richen Liu, Genlin Ji, and Mingjun Su. Domain-specific visualization system based on automatic multiseed recommendations: Extracting stratigraphic structures. *Software Practice & Experience*, 50(2):98–115, 2019.