Proposal of The implement of Creating Evenly-Spaced Streamlines

Fanghai Ge*
Oregon State University

Mingzhao Liu*
Oregon State University

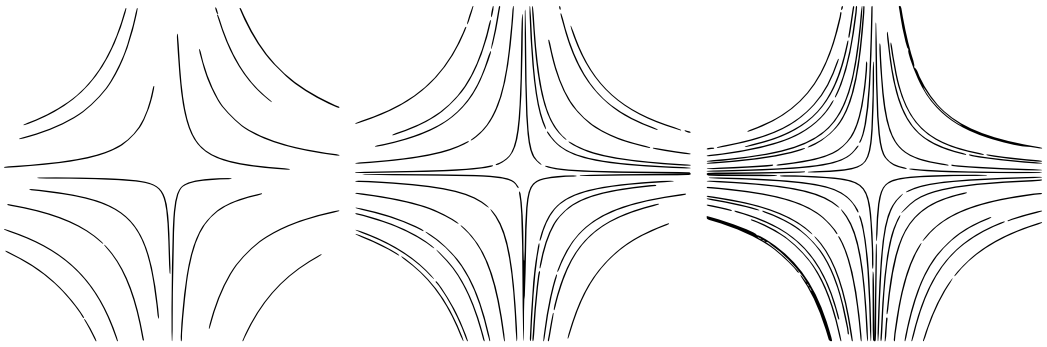


Figure 1: Image comparisons for separating distances of 6%, 3% and 1.5% of image width with streamline placement: (a) 6% separating distance Image, (b) 3% separating distance Image, (c)1.5% separating distance Image. The above image is a field image generated by the final project from Fanghai Ge and Mingzhao Liu. Notice that as the separating distances change, the density of the streamline in the image changes, and the density and the density of the streamline are inversely related.

Abstract

From the related research, the main methods currently used are the LIC method and the IBFV method. The images generated by these two methods are high spatial resolution techniques to properly render ne-grain details, and the amount of calculation is relatively large.

Jobard [1997] proposed the Creating Evenly-Spaced Streamlines method to calculate various flow field images, using seed points to generate streamlines instead of generating all streamline to reduce computational cost. The algorithm can form flow field images of different densities as shown in Figure 1, and the experimental data shows that iterative calculation of streamline has less computational cost and easier rendering process.

Therefore, the main purpose of this final project is to implement the Creating Evenly-Spaced Streamlines algorithm through OpenGL and C++ to visualize the vector field, and to evaluate the implementation results to ensure its correctness.

In order to verify whether the algorithm is completed or not, it is necessary to observe the streamline density of the flow field image. If the streamline density is negatively correlated with the separation distance, the algorithm is successfully implemented. At the same time, in order to verify the efficiency of the algorithm, we will test its running time and compare it with the IBFV algorithm implemented in project 3.The running time of the algorithm should be slightly smaller than the IBFV algorithm.

*e-mail: {gef|liuming}@oregonstate.edu

CR Categories: Final project [CS 553]: The implement of Creating Evenly-Spaced Streamlines;

Keywords: Evenly-Spaced Streamlines, Vector field, Field analysis, Separation distance, Seed points.

1 Introduction

The problem of visualizing vector fields has been widely addressed in the past years because it has numerous applications. The existing flow field visualization algorithms, including the IBFV algorithm and the LIC algorithm, can meet the needs of flow field visualization. However, the main issue of the above algorithm is not suitable for the generation of dense field images and it is difficult to control the generated image. For example, in the IBFV algorithm, even if the mixing ratio of noise can be changed, it is difficult to control the complex features of the image. In the LIC algorithm, it is necessary to filter out unwanted noise through a specific low-frequency filter, which is too complicated for the user.

The method we are currently looking for needs to be able to produce an image of the hand-drawn style and to restore the feature of the flow field well. Therefore, in this final project, we will implement the algorithm of creating evenly spaced streamlines and generating images from hand-drawn styles to LIC styles. Jobard [1997]

The main principle of the algorithm is to generate multiple streamlines until the user-set density level is obtained. After obtaining the first streamline, select a new seed point with the shortest distance from the existing streamline and start

generating the streamline until the streamline exceeds the image boundary. The algorithm ends when no more valid seed point can be found.

The three following sections detail three important points of the algorithm: the control of the distance between adjacent streamlines, seed points selection and streamline integration.

To achieve good visual results, the algorithm uses d-sepration to represent the distance between two adjacent streamlines to facilitate local control of texture generation. In order to speed up the calculation, a uniform sampling point is set on the streamline to calculate the distance from the sampling point to the seed point. In the process of generating seed points, new seed points can only become valid points and join the drawing queue only when their distance is greater than d-separation. In the process of generating streamlines, the new streamline should not be added to the drawing queue. When the distance between the sampling point and the original stream is less than d-separation, the algorithm should only draw the effective streamline pair that satisfies the condition, that is, the minimum distance of the effective streamline pair. Should be greater than 0.5 times d-separation. Inspired by project 3 and 4, we should use the mesh grid to determine the position of adjacent sampling points, and the algorithm can introduce interpolation and approximation to further improve the calculation speed.

In this final project, we make the following contributions.

1. Show which method is used to select sampling points for streamlines
2. Show how to obtain a valid seed point by detecting the distance from the seed point to the sampling point
3. Show how to obtain an effective streamline pair by detecting the distance between the sampling points.
4. Show how to obtain the hand-drawing style or LIC-like style image by reasonably setting the value of d-separation.
5. Test the running time of the algorithm and compare it with the existing LIC algorithm and IBFV algorithm.

2 Division of Tasks

FanghaiGe is responsible for the code of the cell detection part and the sample point pick part, Mingzhao Liu is responsible for the code of the streamline generation part and the parameter adjustment part, and we complete the paper together. The hardware test part uses a home computer with a unified I7-7700K CPU.

3 Agorithm Introduction

For the data set used this time, our main goal is to select seed points with suitable distances near the center of the field source, and then use the seed points to gradually generate streamlines with appropriate distances until the streamlines meet the user-controlled density or there are no suitable seed points. After the seed points are selected, the condition that the streamline stops growing is when the streamline exceeds the boundary or is too close to the nearby streamline.

Pseudocode
Compute an initial streamline and put it into the queue Let this initial streamline be the current streamline

```
Finished := False
Repeat
  Repeat
    Select a candidate seedpoint at d = dsep apart from the
    current streamline
  Until the candidate is valid or there is no more available
  candidate
  If a valid candidate has been selected Then
    Compute a new streamline and put it into the queue
  Else
    If there is no more available streamline in the queue
    Then
    Finished := True
    Else
    Let the next streamline in the queue be the current
    streamline
    EndIf
  EndIf
Until Finished=True
```

3.1 D-sep Concept Introduction

Image density refers to the density of the streamline. When we use a parameter to represent the density of the streamline, you can use the inverse ratio of density to control the density so that it is easy to express.
The image generated in paper has an important characteristic, that is, d-sep, usually d-sep is the distance between the curves. You need to use this parameter to control the distance between the streamline pairs and the termination of the streamline generation process. Image density is inversely proportional to d-sep. Generally, the larger the d-sep, the smaller the image density. We think that d-sep is a suitable distance between line pairs, and the appropriate distance is not the minimum distance of the entire curve, because the streamline is more dense at the field source. If you still use d-sep to generate a nearby streamline, it should cause it to be dense. Streamline pairs are too sparse, which will cause the streamline to be incoherent and this is not suitable for the characteristics of the exhibition venue.
Therefore, it is necessary to increase the d-test scale parameter d-test to control the minimum distance between streamlines. Therefore, the generation principle of the line pair should be the distance between the seed point and the sample point as d-sep, which is an appropriate distance. The condition for stopping the growth of the streamline is that the distance from another streamline is less than d-test or the streamline reaches the image boundary.

3.2 Sample point choose method

We define the sample points during the formation of the stremline. The currently adopted scheme is to set the points used to draw the curve as candidate points. Because the distance of step in single-step operation is less than d-sep.
The separation distance of sample points should usually be 80% of d-sep. Too small separation distance will increase the useless overhead, and too large will cause the detection of seed points to fail. After experiments, usually 80% of the ratio can meet the needs of this data set, and the steps in the stremline generation algorithm we use can be used as the interval of the sample points. We will adjust the ratio of this step to d-sep to obtain high Quality image.

3.3 Streamline Integration Method

The sampling point acquisition method has been introduced in the previous section. The generation of streamline uses two parts, forward and backward. Because the velocity directions of the points on the streamline are the same, when the seed point is not the starting point, if only one velocity direction is used to generate the streamline, the complete line cannot be generated. The seed point is more often located in the middle of the streamline, so half of the streamline can be generated in the original velocity direction. When we reverse the speed direction of the seed point, the other half of the streamline can be generated using the same function. It can be seen that if you want to generate a complete streamline, you need to give the seed point two speed directions and generate a partial streamline. Use getdp and advect when generating (Function model)

3.4 Sedd Point Choose Method

Draw a streamline at the beginning, and select samplepoints on this streamline. The distance between each samplepoints on the same line is less than dsep. The distance between samplepoints between different lines is greater than dtest. When the appropriate samplepoints have been selected, candidate seedpoints with a twice desp distance around the samplepoints are filtered, and the distance between the seed point and the samplepoints is calculated to be equal to dsep. In addition, you need to check that the distance between this seedpoint and the surrounding samplepoint is greater than dtest. At this point, the candidate seed point is the point at which the streamline needs to be drawn. While drawing the streamline, continue to select the samplepoint on this streamline. Continue to find other seedpoints of the initial streamline. When the seedpoint does not exist, continue to find the seedpoint of the next streamline. Finalize the hand-dry evenly treamlines graph.

3.5 D-sep Calculation Method

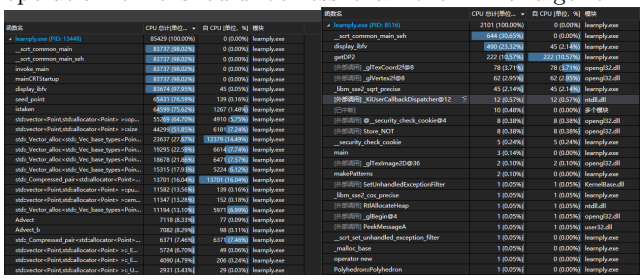
As a local control variable, d-sep not only controls the process of finding the seed point, but also controls the generation process of the streamline. In the streamline generation process, increasing d-sep will increase the success rate of the seed point finding process, but when the size of the seed point is increased to a certain extent, it will affect the efficiency of the algorithm because the judgment process is increased. The value of d-sep is too small, which will reduce the number of seed points, but the data set uses a mesh vertex distribution model, so when d-sep is reduced to a certain degree, the drawing process cannot be completed.
After algorithm analysis, d-sep has the following relationship with the calculation time. When d-sep changes within the range that can be completed, the larger the d-sep, the shorter the seed point search time, the longer the drawing time, and the more d-sep Small, the longer the seed point search time, the shorter the drawing time.

4 Comparation

The IBFV method can accurately observe the velocity and flow direction in the field, and display the field characteristics more completely. At the same time, the real-time control method previously implemented can control the generation of new field sources during the operation of the IBFV algorithm. However, the IBFV function implemented in the job is only limited to simple data sets constructed using mesh. If the multi-field source drawing function is used, when the number of field sources is greater than 4, the running speed is significantly slower, and the use of the OpenGL method to calculate the matrix overhead can easily lead to excessive The machine is stuck, and this is already using a more advanced CPU. The uniformly placed streamline method has a very small overhead, does not require multi-frame synthesis when drawing, and has a huge advantage in terms of computation. It is significantly better than IBFV in the processing of complex data, and the functions that can be completed can show the characteristics of vector fields . In summary, in the case of small data sets, using the IBFV method can better obtain detailed characteristics, such as the specific data concentrated in the mesh, and the memory overhead of processing the data is small. However, when processing large data sets, if we pay more attention to the overall characteristics of the field rather than the characteristics of a single region, such as the streamline distribution shape, the streamline method is obviously better. Therefore, it is not the pros and cons of specific algorithms that are important for data visualization, but the selection of specific algorithms to visualize data based on purpose.

5 Result and Evaluation

In the hardware test section, the test machine is an I7 CPU, and the data set uses the same data set. Compared with the memory consumption of the IBFV mode, usually IBFV uses a multi-frame synthesis algorithm to superimpose the images formed by each frame to finally obtain a flowing field, but the calculation amount and memory consumption are too large. The predicted memory consumption should be greater than the IBFV algorithm, but the overall operation time should be less than the IBFV algorithm.



(a) (b)

Picture a is the CPU operation of our algorithm, and picture b is the CPU operation of the IBFV algorithm. In contrast to the IBFV algorithm, which uniformly generates streamlines for data points in the field, the algorithm requires a secondary screening of the vector field data, and the filtering mechanism is not described in detail in the article. It is speculated that the reason for the excessive memory consumption is that our seed point detection mechanism consumes excessive computing resources, and the algorithm used in the original text should be optimized for the calculation amount of the cell. The consumption rate of our algorithm increases with the increase of seed points. Through the process of generating images, we can also find that the initial generation speed is faster.
Another reason is that we use mesh vertices as data points. This data point as field source data usually makes d-sep detection difficult because the distance of the mesh is constant. In order to fit this data point, a corresponding adjustment needs to be made to d-sep. As mentioned above, under this adjustment, the amount of calculation

usually increases exponentially.
As shown in fig1.a, fig1.b and fig1.c, we have completed changing the density of the image streamline by controlling d-sep, and basically completed the uniform placement of the streamline. But sometimes there are overly dense overlapping lines, and we speculate that the line is caused by some special data points being counted twice into the seed points list. So our next work needs to re-screen the seed points list.

6 Further work

The data set used in this experiment is a relatively simple saddle vector field, and the interaction between the fields has not been detected, even though we have achieved mixed rendering of multiple field sources in previous jobs. Because multiple signal sources may form new signal points after mixing, drawing a streamline around the new signal points may pass the dividing line and then affect the iterative process. This is a possible error we found. So compared to multi-field mixing, the result of a single vector field is predictable and relatively simple. In further work, we may improve the generation method of the streamline, such as using a more elaborate high-order interpolation method for streanline calculation, or improving the data generated by the grid method to make a more accurate drawing. Streamline, through experimental modification, it was found that the experimental data set has a certain effect on the formation of the vector field. If a random data set is used to test the uniformly placed streamline method, the adaptability is poor, so does the control parameter of the streamline need to be increased, and Whether it is necessary to control the area where the streanline is generated through topological methods is of progressive significance.

7 Acknowledgment

Here, I would like to thank our professor Eugene Zhang from CS 553 and our TA Zixuan Feng. In this semester, both the professor and TA have given us great help, and we are extremely honored.

References

Jobard, B., and Lefer, W. 1997. Creating evenly-spaced streamlines of arbitrary density. In Proceedings of the Eurographics Workshop on Visualization in Scientific Computing vol.7, 3, 45–55.