MULTIRESOLUTION FLOW VISUALIZATION

Bruno Jobard and Wilfrid Lefer

Université du Littoral Côte d'Opale
B.P. 719, 62228 Calais, France
e-mail: {jobard,lefer}@lil.univ-littoral.fr

ABSTRACT

Flow visualization has been an active research field for several years and various techniques have been proposed to visualize vector fields, streamlines and textures being the most effective and popular ones. While streamlines are suitable to get rough information on the behavior of the flow, textures depict the flow properties at the pixel level. Depending on the situation the suitable representation could be streamlines or texture. This paper presents a method to compute a sequence of streamline-based images of a vector field with different densities, ranging from sparse to texture-like representations. It is based on an effective streamline placement algorithm and a production scheme that recalls those used in the multiresolution theory. Indeed a streamline defined at level J of the hierarchy is defined for all levels J'>J. A viewer allows us to interactively select the desired density while zooming in and out in a vector field. The density of streamlines in the image can also be automatically computed as a function of a derived quantity, such as velocity or vorticity.

Keywords: Flow Visualization, Streamlines, Multiresolution Representation, Interactive Visualization, Large Vector Fields.

1. INTRODUCTION

Vector field data are produced by scientific experimentations and numerical simulations, which are now widely used to study complex dynamic phenomena, with various areas of applicability, such as global climate modelling and computational fluid dynamics. Large-scale, time-varying simulations are able to produce large amounts of data in a short time and raises the need for effective techniques to get insight in the data and to extract meaningful information. This paper presents a hierarchical approach for the visualization of large two-dimensional steady vector fields.

Several techniques have been proposed to visualize steady flow fields, including icon plots, line representations, and textures. By covering the image with a set of streamlines, the global structure of the vector field can be visualized, its degree of turbulence can be estimated, and the position of all critical points, such as sinks and saddle points, can be easily located in the domain.

To obtain good quality images requires to control the placement of streamlines so as to ensure a uniform density of colored pixels. This issue was addressed by Turk and Banks in [1], and their method was extended to curvilinear grids later [2]. In [3] we have presented a more effective approach, based on a direct placement technique, which uses the separating distance between streamlines to control the density in the image. Here we propose an extension of this technique in order to allow the computation and visualization of streamline images at different levels of density.

The process of exploring a large vector field generally requires different kind of visualization techniques. At the beginning only a rough representation on the whole data domain is needed. As the user refines the region of interest more detailed visualizations are required, to give accurate information on specific parts of the vector field. It should also be possible to visualize different regions with different degrees of accuracy on a single image. So we are interested in how to control the density of streamlines in an image, how to compute a set of images of the same vector field with different densities and hence how to interactively change the desired density of parts of the image while maintaining a given criterion of quality, that is a uniform distribution of streamlines within every part of the image. The main features of the method proposed in this paper are:

- the computation of streamline-based images of a vector field with an optimized visual quality. This is achieved by maximizing the average length of the streamlines used to cover the image domain and by obtaining a uniform density of stream-

lines, i.e. a uniform density of colored pixels in the image. The tapering technique, which has proved to be an efficient factor of enhancement of the quality of streamline-based images, has been easily integrated in our algorithm without any additional computation cost.

- the production of images of an arbitrary density, ranging from sparse representations, that is classical streamline-based images, to dense ones, similarly to those produced by techniques such as Spot Noise [5] and LIC [6][7].
- a strong and efficient control of the density of the image by the user. The density is expressed as the distance between adjacent streamlines, which is actually a criterion that can be easily determined visually, thus it is quite easy for the user to parameterize our algorithm.
- a representation of streamline-based images at different level of density, an algorithm to produce such a hierarchical representation, and a technique to interactively select the desired level of density, for instance during the process of interactive exploration of large vector fields.
- low computation times. The principle of our algorithm is to place each streamline directly at its final place in the image, so that there is no superfluous computation. Indeed we prove that the computation terminates and the computation times for a complete hierarchy of streamlines is those of a single image of the highest density.

The remainder of this paper is organized as follows. Section 2 gives the algorithm of our streamline placement algorithm. Section 3 presents its extension to produce representations of streamline-based images at different levels of density. Section 4 shows how to use such representations for the interactive exploration of large vector fileds and section 5 allows us to conclude.

2. STREAMLINE PLACEMENT

Our streamline placement technique is described in [3]. Here we just give the main algorithm, which is now part of our multiresolution method. The algorithm uses a FIFO list F, which contains the streamlines that have been placed in the image but not yet used for generating new ones. The algorithm generally starts by randomly selecting an initial seed point, from which a first streamline is integrated and put into F. But F can be initialized with any set of streamlines as well, which will be useful for generating images of different densities. In the main loop of the algorithm, a streamline is extracted from F and used for generating new streamlines at its neighborhood. The *IntegrateStreamline()* function computes a new streamline from a given seed point. A streamline is considered as valid if it has the required minimal length. After a new valid streamline has been computed it is necessary to update the list of seed points so as to delete the seed points that are no longer valid. The algorithm finishes when F is empty. This method ensures that we obtain a complete and uniform coverage of the image, thus information about the vector field is available at any point.
By setting the separating distance appropriately, we are able to produce both sparse and dense (texture-like) visualizations with a unified streamline-based approach.

```
Function PlaceStreamlines (V, L_init, d_sep): L_placed

    in:  V : vector field
         L_init : initial streamlines set
         d_sep : separating distance
    out: L_placed : final streamlines set

L_placed := L_init
Foreach streamline S in L_init Do
  Push(F, S)
EndForeach
d_seed := d_sep * ratio_seed/sep
While F is not empty Do
  S_cur := Pop(F)
  L_sp := seed points collection at d_seed from S_cur
  Foreach seed point P in L_sp Do
    S_new := IntegrateStreamline(V, L_placed, p, d_seed)
    If S_new is valid Then
      Push(F, S_new)
      L_placed := L_placed + S_new
      Update L_sp
    EndIf
  EndForeach
EndWhile
Return L_placed
```

3. MULTIRESOLUTION STREAMLINE SETS

There are at least two situations in which a multiresolution streamlines set could be of great interest: enrichment and zoom. Enrichment consists in getting more details from a specific area of interest by adding more streamlines in this region. Zooming means the ability for the user to zoom in and out while maintaining the same density in the image. These two important features in the context of interactive exploration of vector fields can be supported through a multiresolution representation of streamline-based images.

The multiresolution theory is based on the concept of nested spaces, a direct consequence of this being the property that a vector defined at a given level J of the hierarchy is defined for all levels J'>J too. Although there is no such mathematical theory behind our multiresolution representation, this last feature is part of our method, that is a streamline defined at level J is defined for all levels J'>J too. This is a direct consequence of our multiresolution streamlines generation method described in section 3.1. The corresponding algorithm is given in section 3.2.

3.1 Generating Multiresolution Streamlines Images

The set of images at different densities is generated from the lowest density to the highest one. A first set

of evenly spaced streamlines at the lowest density is computed, which yields the image at level 0 of the hierarchy. Then level 1, whose density is higher than level 0, is obtained by adding a set of streamlines to the streamlines defined at level 0, as shown by figure 1. Then the process is repeated until the desired density has been reached. Thus, as for the multiresolution theory, a streamline computed at level J exists for all levels J'>J. Now you can understand why the feature of our streamline placement algorithm, which uses an arbitrary initial set of streamlines, is important.

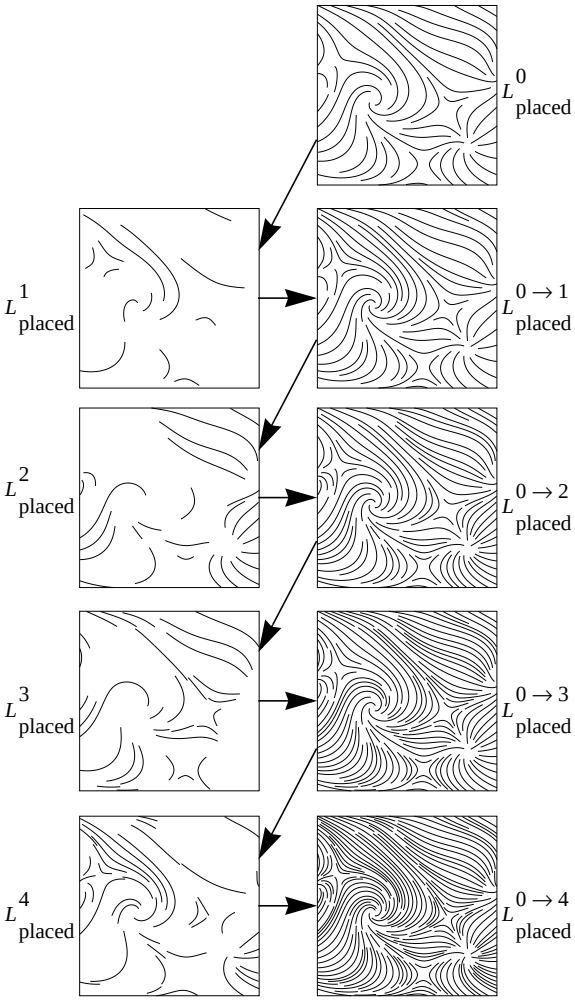

Figure 1: The process of generating multiresolution streamline-based images.

3.2 Algorithm

Our algorithm for generating multidensity images takes as entry parameters the number of levels to be generated and the required densities for the images at the first and last levels of the hierarchy. After the image at level 0 has been computed, the generation of an image for a given level consists in completing the streamlines set of the previous level by new streamlines in order to reach the required density. To do this we use the streamline placement algorithm described in section 2. From one image to the next the separating distance is decreased according to a geometric series, that is $d_{j+1} = d_j$ * *reduction_coef*.
It is possible to start with an arbitrary initial set of streamlines. In this case new streamlines will be added to obtain the image at level 0, then the algorithm processes as explained above. Streamlines in the hierarchy are stored as lists of sample points, they are rasterized only during rendering, which is quite less memory expensive as if all the images at the different levels were stored.

Function **PlaceStreamlinesMultiResolution** (V, L_{init}, min_d_{sep}, max_d_{sep}, J_{max}): L_{placed}

in: V : vector field
L_{init} : initial streamlines set
min_d_{sep} : separating distance at J=0
max_d_{sep} : separating distance at $J=J_{max}$
J_{max} : number of levels minus 1
out: L_{placed} : final streamlines set

reduction_coef := $pow(min\_d_{sep}/max\_d_{sep}, 1.0/J_{max})$
$L_{placed} := L_{init}$
$d_{cur} := max\_d_{sep}$
While $d_{cur} > min\_d_{sep}$ **Do**
  $L_{placed} := PlaceStreamlines(V, L_{placed}, d_{cur})$
  $d_{cur} := d_{cur}$ * reduction_coef
EndWhile
Return L_{placed}

4. INTERACTIVE WALKTHROUGHS IN COMPLEX VECTOR FIELDS

Streamlines are not rasterized in the image as they are computed but rather they are stored as lists of sample points. Thus several rendering modes can be used to make images of these streamlines, depending on which kind of representation is needed, sparse or dense, possibly with glyphs added to the streamlines. Choosing a large distance yields sparse representations, which evoke traditional hand-drawn pictures. Choosing a minimal distance allows to produce texture-like representations, such as those produced by methods like Spot Noise [5] or LIC [6][7].
In order to provide the user with an effective tool for exploring large-scale vector fields, we have developed a multiresolution viewer to navigate interactively within the data. The viewer uses a hierarchy of streamlines sets at different levels of density and allows the user to select the level of details interactively. Moving from level J to the next level J' consists in either adding the streamlines generated at level J' or deleting streamlines generated at level J. Since streamlines are stored as lists of sample points in the hierarchy, drawing or deleting a streamline involves its rasterization in the image. This is not a time-consuming operation and indeed moving from a level to the next is achieved instantaneously. We could also make this operation really smooth by using so-called geomorph functions, whose aim is to make the apparition or disappearance of streamlines as smooth as not to

Figure 2: Three snapshots of an interactive exploration of a vector field using our multiresolution viewer. The suitable level of the hierarchy is selected at any time to maintain a roughly constant density of streamlines.

disturb the observer. In [4] we have proposed such kind of mechanism for handling the apparition and disappearance of streamlines in the context of visualization of time-varying vector fields.

As an enrichment example (see section 3) it is possible to generate an image with varying densities, according to the degree of interest of the different areas or any scalar quantity.

The main problem while zooming on an image is that the amount of details is generally not adapted accordingly. The consequence is that some details are not visible at certain scales and the image lacks details at other scales. Actually the level of details should be evaluated as a function of the image size. Figure 2 illustrates the possibilities our viewer offers for managing the zoom function. The density of streamlines has been set by the user and is maintained continuously during the exploration process, the «best matching» level of the hierarchy being selected at any time.

5. CONCLUSION

A new method has been proposed, which generates streamline-based images at different levels of density. The streamlines are stored as lists of sample points and the right set of streamlines is selected for rasterization as a function of the desired density, which can be computed automatically, as a function of a derived quantity, or set by the user. Thus it is possible to zoom in and out in a complex vector field while maintaining a constant density of streamlines in the image. The placement technique used by our method has proved to be very effective, computing a dense image being achieved in a couple of seconds on any today processor (see [3] for computation times). Let us remark that the number of streamlines in a whole hierarchy is the same as for an image at the highest density, only the organization of the streamlines differs. Hence the time necessary to produce the hierarchy is a couple of seconds.

In certain situations it is useful to compute only a subset of streamlines and to visualize them immediately. For instance, at the beginning of an exploration process of a large vector field, when the user try to get insight in the data and to identify regions of interest, only a sparse image of the whole domain is required. Since our method computes streamlines at increasing density levels, it perfectly fits the requirements of interactive visualization of large data sets.

REFERENCES

[1] Turk, G. and D. Banks. Image-Guided Streamline Placement. Computer Graphics Annual Conference Series (*Proceedings of SIGGRAPH'96*, August 4-9, 1996), pages 453-460, ACM Press.

[2] Mao, X., Y. Hatanaka, H. Higashida and A. Imamiya. Image-Guided Streamline Placement on Curvilinear Grid Surfaces. In D. Ebert, H. Rushmeier and H. Hagen, Editors (*Proceedings of IEEE Visualization'98*, October 18-23, 1998), pages 135-142, IEEE Press.

[3] Jobard, B. and W. Lefer. Creating Evenly-Spaced Streamlines of Arbitrary Density. In W. Lefer and M. Grave, Editors, Visualization in Scientific Computing'97 (*Proceedings of the 8th Eurographics Workshop on Visualization in Scientific Computing'97*, April 28-30, 1997), pages 43-55, Springer-Wien-New-York.

[4] Jobard, B. and W. Lefer. Unsteady Flow Visualization by Animating Evenly-Spaced Streamlines, *Eurographic'00*, August 20-25, 2000.

[5] van Wijk, J. Spot Noise-Texture Synthesis for Data Visualization. Computer Graphics 25(4) (*Proccedings of SIGGRAPH'91*, July 28 - August 2, 1991), pages 309-318, ACM Press.

[6] Cabral, B. and L. Leedom. Imaging Vector Fields Using Line Convolution. In Computer Graphics (*Proceedings of SIGGRAPH'93)*, pages 263-272. ACM Press.

[7] Stalling, D. and H-C. Hege. Fast and Resolution Independant Line Integral Convolution. Computer Graphics Annual Conference Series (*Proceedings of SIGGRAPH'95*, August 6-11, 1995), pages 249-258, ACM Press.