VISUALIZING UNSTEADY FLOWS BY ADAPTIVE STREAKLINES

A. Sanna† **B. Montrucchio**† **R. Arina**‡

†Dipartimento di Automatica e Informatica,
‡Dipartimento di Ingegneria Aeronautica e Spaziale,
Politecnico di Torino
corso Duca degli Abruzzi 24,
10129 Torino (Italy)
email: sanna@polito.it

ABSTRACT

The visualization of unsteady flows is an attractive field of research and texture-based techniques seem to provide satisfactory results. In this paper we propose a texture-based method which follows streaklines in order to produce an effective visualization of time-dependent phenomena. The local vorticity of the field allows to trace a number of streaklines according to the flow characteristics; in this way, more significant areas are denoted by a higher level of detail. Moreover, different colors are assigned to different vorticity levels to better denote flow instabilities.

Keywords: Scientific visualization, unsteady flows, adaptive particle tracing, level of detail.

1 INTRODUCTION

Vector field data arise from computer simulations in a variety of disciplines, such as CFD (Computational Fluid Dynamics), climate modeling and so on. An adequate visualization is a challenging problem due to the difficulties in finding a suitable graphic way to represent and display vector data on a monitor display.

Of particular interest is the visualization of time-dependent phenomena; in these cases, vector data are produced for a set of time steps. To visualize time-varying data, two kinds of approaches can be employed: *instantaneous methods* and *time-correlated methods*. By the first approach, each time step is considered separately from the others and a graphical icon representing the vector field is produced. The discrete time steps are animated together to obtain the resulting visualization sequence. *Time-correlated methods* compute a graphical icon for each frame considering the correlation of the current time step with the previous ones.

Instantaneous methods are mainly based on streamlines which represent field lines envelopes of the tangents to the vector field at a given time level. Unfortunately, these techniques often suffer of lacking of temporal coherence between frames since the streamlines shown at a given instance in time do not correspond to the paths that the particles have traveled inside an unsteady flow field.

Time-correlated methods can overcome this problem. In this paper we propose a technique which follows particle traces (in particular streaklines) in order to produce a texture for each time step of an animation. A different number of streaklines is traced according to the local vorticity of the flow field; a larger number of traces is displayed for the areas where higher velocity gradients exist, while a smaller number of streaklines is traced where the flow is smoother. Moreover, each particle of a streakline is denoted by a color depending on the local vorticity intensity of the area where the particle is placed.

A set of insertion points (an insertion point is a location from where the particles are released) will correspond to a set of pixels on the output texture. The insertion points are kept constant for all time steps, and at each step a new particle is released from a location. In the meanwhile, the positions of the previously emitted particles are updated. In this way, each frame of the animation maintains the coherence with the previous ones and the resulting sequence can effectively show the evolution of the field in time.

In order to guarantee a minimum level of details in each area of the resulting image, the streaklines starting from a subset of insertion points are traced independently of the vorticity values. On the other hand, the particles released from the other insertion points will affect the output texture only in the event they are placed in field zones where the vorticity is greater than a threshold set from the user. In this way, the user can tune the magnitude of the flow field details to be displayed.

The paper is organized as follows: main visualization techniques are reviewed in section 2, goals and ideas of the proposed paper are outlined in section 3, and the algorithm is shown in the details in section 4. Finally, two examples are presented in section 5 where we propose a comparison between the proposed method and a technique following non adaptive streaklines [Sanna99b].

2 BACKGROUND

An effective way to visualize unsteady flows is to compute particle traces [Niels97a]. In steady flows streamlines represent field lines envelopes of the tangents to the vector field at a given time level. For unsteady flows, which are of interest for this paper, three types of particle traces can be computed: *pathlines* (show the trajectory of a single particle released from an insertion point), *streaklines* (lines joining the positions, at an instant in time, of all particles released from the same insertion point), and *timelines* (lines joining the positions of particles released at the same instant in time from different insertion points). In an instantaneous flow field, streamlines, pathlines, and streaklines are identical [Saffm92a] but it has been proved that streaklines and timelines can visualize information not revealable by pathlines for unsteady flows: [Niels97a], [Lane93a], and [Lane96a]. For instance, in [Niels97a], an example shows as vortices behind an airfoil are visibile by streaklines but streamlines and pathlines are not able to denote these characteristics clearly. Unfortunately, particle tracing techniques depend critically on the placement of the insertion points; depending on their placement details and structures in the flow field may be missed.

Another approach for visualizing vector field data is based on texture synthesis. In particular, the LIC (Line Integral Convolution) technique revealed to be an effective and elegant method to visualize vector field data [Cabra93a]. The LIC method takes a vector field and a white noise image as the input; the algorithm performs one-dimensional convolution on the noise image. The convolution kernel follows the paths of streamlines originating from each pixel in both positive and negative directions. The resulting intensity values of the LIC pixels along each streamline are strongly correlated, and hence the directional patterns of the flow fields can be easily visualized. The LIC algorithm as presented in [Cabra93a] is applicable only to vector fields over regular 2D Cartesian grids; moreover, the LIC shows only the direction of the flow but not its orientation and it is not adequate to visualize animation of unsteady flows. In order to solve these drawbacks several works have been published in the literature, among these: fastLIC [Stall95a] allows to speed up the LIC of an order of magnitude, OLIC (Oriented Line Integral Convolution) [Wegen97a] was designed to show the orientation of a flow by sparse textures, a 3D LIC was presented in [Inter98a], a fast algorithm to display direction, orientation and magnitude by dense textures was presented in [Sanna99a], furlike textures [Khous98a] are used in [Khous99a], and a parallel convolution algorithm based on pathlines was proposed in [Shen98a].

In particular, in [Shen98a] has been outlined as all the previous techniques which follow traces of streamlines cannot be effectively employed to visualize unsteady flows since the approach of computing the LIC (or an its evolution) for each time step and then animating the results together may suffer of lacking temporal coherence between animation frames.

3 GOALS AND BASIC IDEAS

The problem of instantaneous methods can be tackled following traces of pathlines as in [Shen98a] or, more effectively, traces of streaklines as presented in [Sanna99b]. In this paper we propose an improvement to [Shen98a] and [Sanna99b]. It has been experimentally proved that streaklines can reveal flow characteristics otherwise not detectable by streamlines and pathlines when time-dependent phenomena are investigated. On the other hand, graphical icons resulting following streaklines may sometimes confuse the user. In particular, streaklines can be confused easier than streamlines and the streakline orientation may be not clear if a frame of the sequence is analyzed singularly [Sanna99b]. The analysis of a single frame does not allow to completely evaluate the differences between streaklines and streamlines since time coherence between frames cannot be appreciated, but significant differences can be noted. In Fig. 1 a frame (belonging to a sequence of 84 time steps) is shown for two techniques: the left image has been produced by following non adaptive streaklines while the right one by an instantaneous method. Although the instantaneous method reveals the two flow instabilities (areas 2 and 3) it cannot well depict the vorticity of the area 1. Although streaklines allow often to visualize characteristics not otherwise reveleable and the entire animation does not suffer of lack of coherence showing magnitude and orientation of the flow, the investigation of a single frame can be difficult.

Our goal is to overcome this problem allowing a better understanding of flow characteristics even if only a single frame can be analyzed. The problem is mainly due to the fact only in some areas should be necessary a high level of detail, while elsewhere a lower number of traces is sufficient. To tackle this issue, the study of

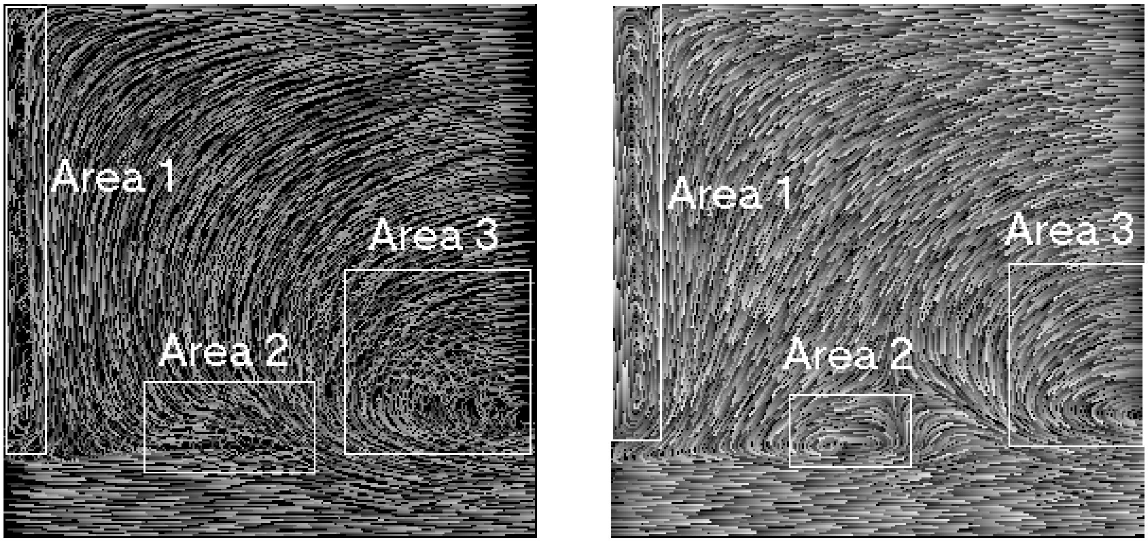

Figure 1: A comparison between streaklines and streamlines.

the vector field can help us to detect the areas where flow field contains non trivial structures and where a larger number of streaklines is required.

The motion of a fluid is described by a vector field $\mathbf{V}(\mathbf{x}, t)$, representing the velocity of the fluid. The curl of the velocity is termed vorticity $\omega(\mathbf{x}, t)$. The importance of the vorticity for the description and understanding of fluid flows stems from its kinematical interpretation as twice the angular velocity around an infinitesimal circle. Regions of fluid in which the vorticity is identically zero, are said to be irrotational, and fluid elements are carried along with the flow without rotation. On the contrary in the rotational regions the fluid elements are constrained to rotate and are subjected to intense strains leading to complex flow structures.

For two-dimensional flows the vorticity vector has only one component along the direction perpendicular to the plane of the motion, given by the relation

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \qquad ,$$

where (u, v) are the velocity components along the coordinate directions (x, y).

In the examples under study, the main phenomenon of interest is the mixing of incompressible fluids between two regions characterized by a different level of vorticity. The vorticity is a measure of the rotational movement of a fluid particle, and in our cases is the only suitable indicator for locally tracking the mixing process. The analysis of the local vorticity allows us to adapt the number of streaklines to local flow characteristics; moreover, different colors can be assigned to particles according to vorticity values in order to effectively visualize the vortical regions.

4 THE ALGORITHM

Since streaklines can better characterize the evolution of flow fields in time, the algorithm has to be able to depict the particle traces as a texture. Moreover, a different level of the detail must be chosen according to the flow field characteristics. The local vorticity (see section 3) allows to denote the areas of the field where a larger number of streaklines has to be depicted as well as the zones where a lower density of traces is sufficient.

In order to do this, a set of insertion points must be selected (denoted as `ins_pts`); since we produce a texture for each time step of the flow field evolution, the insertion points will correspond to certain pixels of the output texture. The choice of the seed locations must be as much as possible uniform in order to inspect each part of the field. A random positioning of these locations can be obtained, for instance, by a quasi-random Sobol distribution. At each time step a particle has to be released from each insertion point; the locations of the insertion points are unchanged for all time steps since the Sobol distribution is re-initialized.

Particle traces affect the output texture only where the local vorticity is greater than a threshold `T` set from the user (and constant for all frames of the animation). In order to avoid that areas of the output texture lack of any trace, the user can set a percentage of insertion points (denoted as `fix_pts`). The streaklines starting from the `fix_pts` will be displayed on the output texture independently of the local vorticities. This allows a minimum coverage, and hence a minimum level of detail for the whole image.

The length of the streaklines (`L`) can be set from the user (and constant for all frames of the animation) at the beginning of a visualization. For each time step a new particle is released from an insertion point. If the streakline starting from an insertion point is longer than `L`, that is at least `L+1` particles have been emitted, the “oldest” particle (the particle `L+1` old) is dropped. Afterwards, the positions of the particles previously emitted are updated according their locations inside the vector field. This means that, after an initial transitory, if the number of insertion points is denoted by `N` and the streakline length is `L`, the complexity of the algorithm is `O(N·L)`.

When a particle goes outside the image borders, i.e. exits from the vector field, it is marked as an invalid particle and its position is not more considered.

After updating the positions of all particles, the pixels of the output texture affected by each streakline must be computed. The color of a pixel affected from a particle depends on the local vorticity. In order to assign a color, we map the vorticity values as they were the component `H` of the `HLS` color system where `L` and `S` are fixed and only `H` can change. Blue tones correspond to low vorticities, green tones to medium vorticities, and red tones to high vorticities; in order to obtain a smooth transition between different level of vorticities, a logarithmic scale is adopted.

Each particle released from the first set of points (the `fix_pts`) can affect a pixel of the output texture, independently of the vorticity values (i.e. a color is assigned even if the local vorticity is lower than the threshold); on the other hand, only when the local vorticity is greater than the threshold a color is assigned to a pixel affected from particles belonging to streaklines starting from the other insertion points (the `ins_pts` that are not `fix_pts`). In this way, the level of detail grows where the field is most significant.

A simple two-dimensional array can be used in order to effectively manage the particles. The index of row denotes the insertion points while the index of column indicates the time steps. Each element of the array can be a record which stores the information about a particle released at a time step. When the number of particles released from an insertion point is equal to the streakline length the index of column is set to zero, in this way the information about the new particle will overwrite the oldest one.

It is worth to denote that the streaklines are computed for all insertion points, but a set of them is visualized only where a higher level of detail is required from field characteristics. Another approach could dynamically vary the number of insertion points according to the local vorticity values. This could be effective for a slow evolution of a vector field since there is a delay in frames (equal to the streakline length) to fully compute the streaklines. This drawback could not be negligible for highly unsteady flows.

5 EXAMPLES AND RESULTS

The frame shown in Fig. 1 has been computed also for adaptive streaklines and it is reported in Fig. 2; the important regions are more effectively denoted respect to streamlines and non adaptive streaklines.

Further we present other two examples. In the first example, we visualize the flow field corresponding to the natural development of a spatially evolving two-dimensional laminar mixing layer. A mixing layer originates in the merge of two parallel streams, each with a uniform velocity U_1 and U_2 ($U_1 > U_2$), both assumed in the same direction. During its development in the streamwise direction, the vorticity, initially possessed by the layer, undergoes to a redistribution in space. This evolution is made up of two distinct phases. During the first phase, the mixing layer rolls up into periodic structures with a characteristic length of the same order of the mixing layer thickness. The second phase is a result of a further instability leading to a pairing of the periodic structures, with a dramatic increase of the mixing layer thickness. The two-dimensional flow field has been computed by numerical simulation of the Navier-Stokes equations for an incompressible fluid. The resulting flow field is unsteady and characterized by coherent structures of concentrated vorticity.

A frame (of a sequence of eighty time steps) is shown in Fig. 3; the left picture is obtained by non adaptive streaklines, on the other hands, the right figure by the proposed methodology. The adaptive streakline visualization significantly enhances the details of the complex vortical structure inside the shear layer, pointing out the presence of travelling vortex cores and the regions of entrained fluid. The number of `ins_pts` is equal to 3% of the total (Xres*Yres) by using adaptive streaklines, while the number of `fix_pts` is equal to 10% of the `ins_pts`. For non adaptive streaklines, the number of `ins_pts` has been decreased to 1% in order to avoid an excessive density of traces in areas where local vorticity values are low.

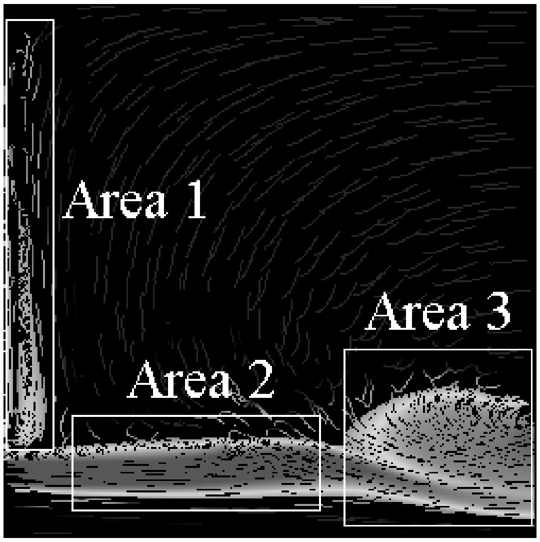

Figure 2: The example shown in Fig. 1 computed by adaptive streaklines.

In the second example, we visualize the flow field past a backward facing step. The typical flow pattern displays the formation of a separated flow past the step edge, as well as the emergence of reattached flow downstream, on the lower wall. For an incoming laminar boundary layer at Reynolds number of 500, the flow field is unsteady: an important part of the recirculating fluid behind the step is periodically shed, and large vortical structures flow downstream along the wall. The

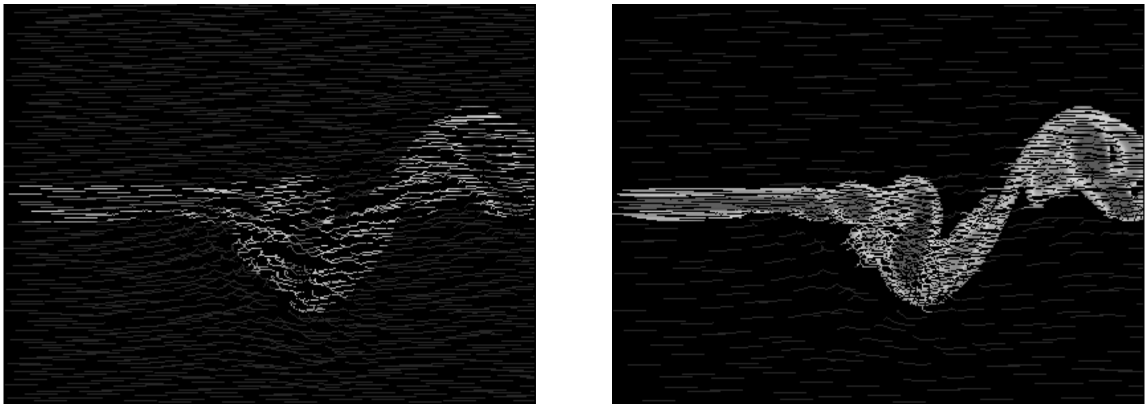

Figure 3: Rolling up of laminar mixing layer.

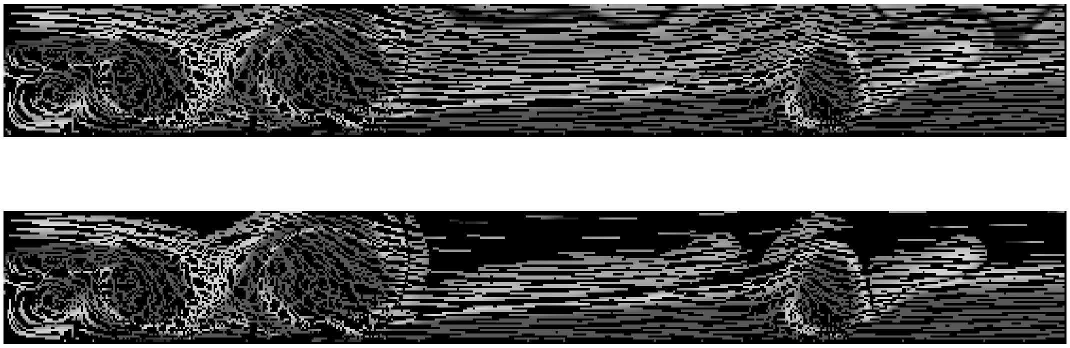

Figure 4: Vortex shedding past a backward facing step (laminar flow, Re=500).

phenomenon is characterized by a complex vortex dynamics which is revealed by the present visualization technique. A frame (of a sequence of 100 time steps) is shown in Fig. 4; the second picture has been obtained by using adaptive streaklines. It is possible to recognize the region formed by the shedding shear layer, due to a Kelvin-Helmholtz instability of the separated boundary layer, from the above almost irrotational fluid. For the second example the `ins_pts` are equal to 3% of the total (both for adaptive and non adaptive streaklines) and the `fix_pts` are 10%. Since the areas having low vorticity values are just a few parts of the image, the two approaches (adaptive and non adaptive) presents a very similar behavior, but adaptive streaklines better characterize instabilities and structures of the flow field.

The two examples have got a resolution respectively of: 402 x 302 and 562 x 62 pixels; both visualizations have been computed on a 433 MHz DEC alpha workstation needing 14.9 seconds for the first example (5.4 fps) and 7.0 seconds for the second case (14.3 fps). The entire visualizations of the proposed cases may be seen at: http://www.hipeco.polito.it/wscg2000/. Unfortunately, optimal values for: the threshold `T`, the lenght `L`, the percentage of insertion and fixed points must be found by a set of tests, and cannot be tuned automatically, yet.

6 CONCLUSION

An algorithm for visualizing unsteady flows has been presented. The proposed approach adapts the number of streaklines traced according the local vorticity values; moreover, the highly vortical regions are denoted by varying the colors again according to the vorticity values. Adaptive streaklines allow to obtain a level of detail dependent on the characteristics to be visualized. This allows a better understanding even if each frame of the animation is analyzed singularly. The proposed method: does not require an input texture, its computational cost can be controlled by the number of insertion points, can be easily implemented, and it does not suffer of lacking of temporal coherence between frames.

Future work will be aimed to: find metric rules to automatically set both threshold value and number of insertion points which are experimentally tuned in this paper, and extend this algorithm for 3D applications.

REFERENCES

[Sanna99b] Sanna, A., Montrucchio, B. and Arina, R.: *On Time-Varying Flow Fields: a streakline-based visualization method*, Eurographics'99 Short Papers and Demos Proceedings, pp. 30-33, 1999.

[Niels97a] Nielson, G.M., Hagen, H. and Müller, H.: *Scientific Visualization*, *IEEE* Computer Society, Los Alamitos, California, 1997.

[Saffm92a] Saffman, P.G.: *Vortex Dynamics*, Cambridge University Press, Cambridge, 1992.

[Lane93a] Lane, D.A.: *Visualization of Time-Dependent Flow Fields*, In *IEEE* Visualization'93 Proceedings, pp. 32-38, 1993.

[Lane96a] Lane, D.A.: *Visualizing Time-Varying Phenomena in Numerical Simulations of Unsteady Flows*, AIAA 96-0048, 34th AIAA Aerospace Sciences Meeting, Reno, Nevada, 1996.

[Cabra93a] Cabral, B. and Leedom, L.C.: *Imaging Vector Fields Using Line Integral Convolution*, ACM Computer Graphics (Proc. of SIGGRAPH '93), Vol. 27, pp. 262-270, 1993.

[Stall95a] Stalling, D. and Hege, H.C.: *Fast and resolution independent line integral convolution*, ACM Computer Graphics (Proc. of SIGGRAPH '95), pp. 249-256, 1995.

[Wegen97a] Wegenkittl, R., Gröller, E. and Purgathofer, W.: *Animating Flowfields: Rendering of Oriented Line Integral Convolution*, In *IEEE* Visualization'97 Proceedings, pp. 15-21, 1997.

[Inter98a] Interrante, V. and Grosch, C.: *Visualizing 3D Flow*, *IEEE* Computer Graphics & Applications, Vol. 18, No. 4, pp. 49-53, 1998.

[Sanna99a] Sanna, A., Montrucchio, B., Arina, R. and Massasso, L.: *A 3D Fluid Flow Visualizer for Entry Level Computers*, In Proceedings of WSCG'99, pp. I249-I256, 1999.

[Khous98a] Khousas, L., Odet C. and Friboulet, D.: *3D furlike texture generation by a 2D autoregressive synthesis*, In Proceedings of WSCG'98, pp. 171-178, 1998.

[Khous99a] Khousas, L., Odet C. and Friboulet, D.: *2D Vector Field Visualization Using Furlike Texture*, In Proceedings of the Joint Eurographics and *IEEE* TCVG Symposium on Visualization, pp. 35-44, 1999.

[Shen98a] Shen, Han-Wei and Kao, D.L.: *A New Line Integral Convolution Algorithm for Visualizing Time-Varying Flow Fields.*, *IEEE* Transaction on Visualization and Computer Graphics, Vol. 4, No. 2, pp. 98-108, 1998.