Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

Spotting Structure in Complex Time Dependent Flow

Wim de Leeuw, Robert van Liere

Software Engineering (SEN)

SEN-R9823 September, 1998

Report SEN-R9823
ISSN 1386-369X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Spotting Structure in Complex Time Dependent Flow

Wim de Leeuw

Robert van Liere
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
Analysing structure in complex time dependent flow fields is a challenging problem. Spot noise is a technique which utilizes texture for the visualization of flow fields. In this paper, two extensions of spot noise are discussed. These extensions allow spot noise to be used for very detailed analysis of time dependent flow fields. Two applications are discussed: a large data set resulting from a direct numerical simulation of turbulence and a data set resulting from a global climate data.

1991 Computing Reviews Classification System: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.6 [Computer Graphics]: Methodology and Techniques I.6.6 [Simulation and Modeling]: Simulation Output Analysis.
Keywords and Phrases: interactive scientific visualization, flow visualization, texture generation.
Note: Presented at Dagstuhl Seminar No. 9724 *Scientific Visualization.* Submitted for publication.
Work carried out under project High performance visualization

1. Introduction.

During the last decade the need for flow visualization techniques to represent vector fields has become apparent. The reason for this is that increasingly complex phenomena are simulated and that the resulting data sets are becoming more difficult to interpret. The scientific visualization community has developed many accurate and robust flow visualization techniques to display these data sets [1]. Simultaneously, there is a growing need in the high performance computing community to apply these techniques to large, time dependent, flow fields. These fields are characterized by complex flow patterns of widely varying spatial and temporal scales.

Spot noise is a texture synthesis technique which can be used to visualize flow fields [2]. Rather than mapping the underlying vector field to produce colored geometric objects, spot noise uses texture. The primary advantage of using texture over other flow visualization techniques is that texture can give a continuous view of a 2D field opposed to visualization at only discrete positions, as with arrow plots or streamlines. Spot noise generates texture by adding a large number of randomly positioned spots with a random intensity. A spot is defined to be any geometric shape, but usually a small circle is used. The idea is that if many spots are blended together in one image, the individual spots can no longer be discerned and texture is perceived instead. Properties of the spot directly control the properties of the texture. Therefore, by modifying the shape of the spot as a function of the data, the data are visualized by texture.

The motivation of the work in this paper is driven by the application of spot noise to very large data sets. During discussions with the applications developers, it quickly became apparent that spot noise lacked two features: the technique was not suited to display detailed views of the underlying data and it was not able to cope with time dependent data. The spot noise technique has been extended in order to cope with these deficiencies. The organization of this paper is as follows: We

first summerize previous work on spot noise. We then then discuss the zooming and time dependent extensions. Finally, we discuss how spot noise has been applied to two very large data sets.

2. Spot noise.

2.1 Overview.

We give a brief overview of the spot noise texture synthesis technique. A texture can be characterized by a scalar function f of position $\mathbf{x}$. A spot noise texture is defined as

$$f(\mathbf{x}) = \sum a_i h(\mathbf{x} - \mathbf{x}_i)$$

in which $h(\mathbf{x})$ is called the spot function. It is a function everywhere zero except for an area that is small compared to the texture size. a_i is a random scaling factor with a zero mean, $\mathbf{x}_i$ is a random position. In non-mathematical terms: spots of random intensity are drawn and blended together on random positions on a plane (figure 1).

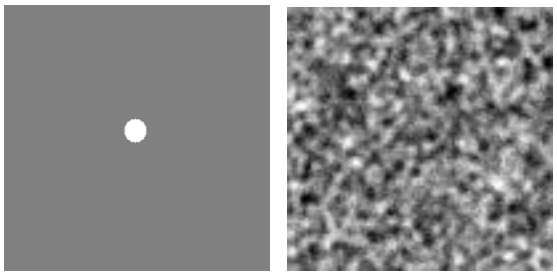

Figure 1: Principle of spot noise: single spot (left) resulting texture (right)

The shape and size of spots determine the characteristics of the texture. By local variation of the spot shape, presentation of data becomes possible. For example, spots can be rotated and scaled to reflect the local flow. First the spot is aligned with the flow direction by a rotation, then it is scaled. In the direction of the flow it is scaled proportionally to $1 + |\mathbf{v}|$ and perpendicular to the flow by $1/(1 + |\mathbf{v}|)$, where $|\mathbf{v}|$ is the velocity magnitude. This scaling ensures that the total area of the spot and thus the average density in the texture remains constant. A zero velocity results in a round spot.

In [3], enhancements to the original spot noise algorithm are given. Bent spots allow spot noise to be used in irregular flows, such as in areas where curvature of the vector field is high. Instead of only scaling and rotation, bent spots are deformed using a mesh to represent the local flow. The mesh is generated by tiling a surface generated by advecting streamlines in the flow.

2.2 Parallel spot noise.

The downside of spot noise is that it is very computationally expensive. Previously, spot noise textures were generated as a preprocessing step and the resulting sequence of images were animated. A scalable parallel algorithm has been developed which allows textures to be generated on the fly. This allows the user to interactively adjust spot noise parameters when generating animations of textures. The parallel algorithm not only exploits multiple processors, but also fully utilizes the available graphics subsystems.

Before discussing a parallel version of the spot noise technique, we introduce the sequential spot noise pipeline. Algorithmically, spot noise synthesis performs four steps (figure 2): First, read a data set of a vector field. Second, the position of each particle is updated by advecting it by the flow field. Third, a spot is generated for each particle position. Each spot is transformed and assigned an intensity according to properties of the flow field. All spots are scan converted and blended together

to form a texture map. Finally, an image is rendered by mapping the texture onto a geometric surface. Other visualization techniques may also be superimposed on the texture mapped object.

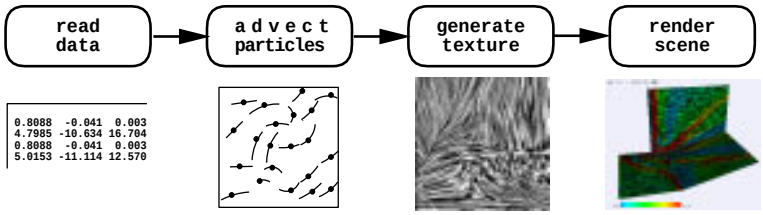

Figure 2: Spot noise pipeline; logical steps (top) and iconic representations (bottom).

The parallel algorithm uses a straight forward divide and conquer strategy, [4]. The underlying idea of is based on the observation that spots are independent and that the amount of work to be performed on a spot is constant. The parallel noise pipeline is illustrated in figure 3. The collection of particles is partitioned into a number of disjunct sets. Each particle set is processed by one or more processors and one graphics pipe. The processing of a particle set results in an texture. After completion, these textures are gathered and blended to form the final spot noise texture.

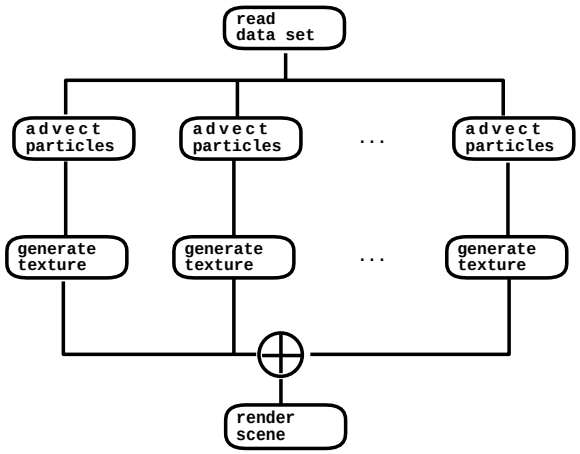

Figure 3: Divide and conquer spot noise pipeline.

Ideally, the total time spent by the divide and conquer algorithm is :

$$T \;=\; \max(\sum_{i=1}^{N} genP_i/nP \;,\; \sum_{i=1}^{N} genT_i/nG) \;+\; c \tag{2.1}$$

in which :

$genP_i$	=	processor time to calculate position and shape of spot i
$genT_i$	=	time to blend spot i into the texture
nP	=	number of processors
nG	=	number of graphics pipes
c	=	overhead costs due to additional texture blending

Equation 2.1 is based on two assumptions. First, in order to move the raw geometric data, there is sufficient bandwidth between the processors and graphics subsystem. Second, in order to avoid under utilization, the data generated by the processors can be streamed to the graphics subsystem.

In [4], it was shown that by using the divide and conquer method, interactive generation of high quality textures can be achieved on a powerful graphics workstation. In addition, by adjusting of spot noise parameters, such as the number and size of spots, speed can be traded for quality and very high generation speeds can be realized.

3. Zooming and time dependent extensions.

3.1 Zooming.

The quality of the final spot noise texture is based on the size and density of the spots. Characteristics of a spot (such as size, shape and direction) directly influence the characteristics of the texture (such as granularity and isotropy). More precisely, the characteristics of a texture are captured by the two dimensional power spectrum $P_f\ (\omega)$ defined by

$$P_f\ (\omega) \ = \ \lim_{T\to\infty} \frac{1}{T}|F_T\ (\omega)|^2 \tag{3.1}$$

in which $F_T\ (\omega)$ is the Fourier transform of a sample of the texture with size T. In [2] it was shown that the power spectrum of the texture is proportional to the power spectrum of the spot.

The data sets we studied in this paper have high resolutions. However, since the spatial resolution of the texture is limited, zooming in on detail of the data will result in large texels in the image on the screen. Increasing the texture resolution would only partially solve this problem. More adequate would be to develop a multi-resolution scheme which allows both global and detailed views of the data set, without compromising on the texture quality on the screen.

A solution to zooming is to ensure that the power spectrum of the texture visible on the screen remains constant during zooming. This can be achieved by adjusting two parameters. First, a region of interest can be defined based on the amount of data visible on the screen. Texture will only be generated for this region. Since the region of interest is smaller, the size of the cells (and therefore the smallest details in the flow) become large relative to the texture and pixels on the screen. Second, the size of the spot can be decreased. In this way, the size of the spot relative to the size of the texture will remain constant.

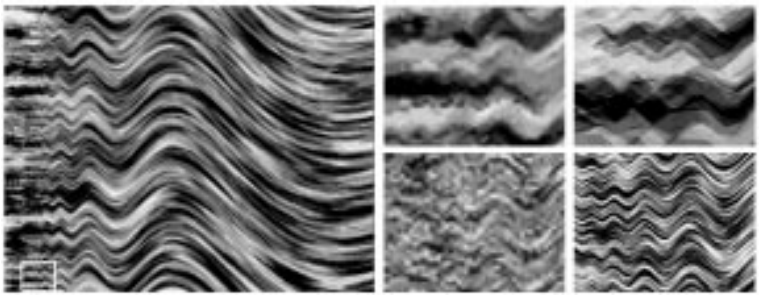

Figure 4: Zooming in a synthetic flow. On the left is a global view of the flow. On the right are four images of the indicated region. From top left to bottom right: zooming without adjustment of spot size or region of interest (large texels), zooming with decreased region of interest (large spots), zooming with decreased spot size (white noise), zooming with decreased spot size and region of interest.

Zooming is illustrated in figure 4. On the left is a global view of a synthetic flow field. The data is defined on a rectilinear grid with a size of 180x180 cells. The resolution of the texture is 512x512. A small bounding box is drawn in the lower left corner which indicates the zooming area. The size of this region of interest is 30x30 cells. Also, four variations of zooming are displayed. The top left image zooms in without adjusting the spot size or region interest, resulting in an image with very large texels. In the top right image, only the region interest has been adjusted, resulting in an image

with very large spots. In the bottom left image, only the spot size has been decreased, resulting in an image which approaches white noise. The high frequencies resulting from the small spot size cannot be represented by such a low resolution texture. Finally, in the bottom right image, both the spot size and the region of interest have been decreased, resulting in a high quality detailed image. Here, the power spectrum of the visible texture is approximately equal to the power spectrum of the texture in the global image.

Manual adapting the spot size or region of interest requires some expertise in which the interactive spot noise algorithm, in which new textures can generated in real time, is instrumental. The automatic adaptation of spot size and region would be complex and is beyond the scope of the current research.

3.2 Time dependent spot noise.

The challenge of texture animation is to maintain the coherence of textures. To perceive flow, two issues must be addressed. First, the spatial coherence in a texture must be maintained; i.e. the intensity variation in a texture should properly reflect the data. Second, temporal coherence between frames must be maintained; i.e. the movement of a spot should properly reflect the flow.

In a stationary flow, temporal coherence is obtained by using a technique based on particles and cyclic display. Between successive frames spots are advected along streamlines. Particle positions on the streamline are calculated by:

$$\mathbf{x} = \mathbf{x}_0 \; + \; \int_{x_0}^{x} \mathbf{v}(\mathbf{x}) \; dx \tag{3.2}$$

where $\mathbf{x}$ is the particle position, and $\mathbf{v}(\mathbf{x})$ is the velocity of the particle. During a lifetime T, each spot is advected over a small distance along the streamline to obtain a new position. The intensity of a spot is modulated between zero at its initial position, a maximum value and zero again at its final position. The length of the lifecycle is equal for all spots, but the phase of the lifecycle at a certain moment varies randomly over all spots. This technique has given excellent results in animating sequences in stationary flow [2].

For time dependent flow, spatial coherence is obtained only if particle paths are used to advect the spots. Positions on a particle path are expressed as:

$$\mathbf{x}(t) = \mathbf{x}(t_0) \; + \; \int_{t_0}^{t} \mathbf{v}(\mathbf{x}(t), t) \; dt \tag{3.3}$$

Spots are regarded as particles on the particle path and their positions are calculated accordingly. However, as spots are advected over time, the distribution of the spots can change rapidly. As such, the spatial coherence of the spots may be compromised. To achieve a uniform spot distribution, the previously mentioned lifecycle mechanism is extended. Spots are advected from set of random positions. When a spot reaches the end of its lifecycle, it will restart its lifecycle at its initial position. A lifecycle of a spot has a typical length of about 15 frames. In this period the average distance covered by the particle is sufficiently small to keep a uniform distribution of spots.

Bent spots were introduced to cope with irregular, highly curved, flow patterns. In the stationary case, streamsurfaces are used to calculate the curvature of a bent spot [3]. However, for time dependent irregular flow additional care must be taken to maintain coherence of spots. Two different approaches can be taken to calculate the curvature of the bent spot. First, particle paths can be used. The texture reflects the temporal change of the flow. Second, streamlines can be used. The texture reflects instances of the flow without taking history or future into account. [1]

We illustrate this in figure 5. The three columns show different time steps of a flow field. The flow is uniform and rotates over time, as illustrated in the top row of the figure. In rows two and three, two spots are traced over time. Stippled lines indicate particle paths and, hence, the paths of the spots.

[1] These coherency issues also exist when the LIC technique is used. This is discussed in [5].

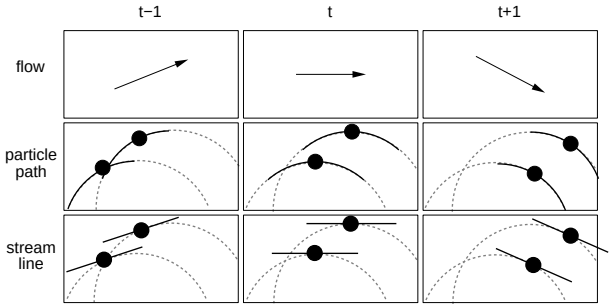

Figure 5: Particle positions over time.

Bold lines indicate the bending of spots. If particle paths are used, the shape of a spot coincides with the path of the spot. If streamlines are used, the shape of a spot does coincides with the flow at a particular time. Note that in this particular case streamlines are straight line segments because the flow is uniform.

While using particle paths to determine the spot curvature maintains temporal coherence, it compromisesz the spatial coherence of the texture. This is because particle paths, and therefore spots, may intersect, resulting in artifacts in the texture (as is shown in the first column of the figure). Using streamlines may compromise the temporal coherence of the texture, as the actual path of the spot may differ from the path suggested by its shape. Both approaches to spot bending have their disadvantages. It is application dependent if the resulting artifacts are noticeable. In the cases we have studied, the problems in temporal coherence using the streamline approach were less severe than the problems of spatial coherence introduced by using the particle path approach.

4. Applications.

4.1 Direct numerical simulation of a turbulent flow.

In [6], methods are discussed for direct numerical simulation of turbulent flow. The goal of this application is to study the evolution of the vortex shedding behind a block. The transition from laminar to turbulent flow is still an ill understood phenomenon. By using spot noise as an interactive data analysis technique, users can quickly test different hypotheses about this phenomenon.

The computation and the size of the resulting data set is impressive: a few weeks of computing can easily produce a few terabytes of data. A data browser is being developed to analyse such scientific data bases. In contrast to prerecorded video sequences, the data browser allows the user to first select visualization mappings and then play through any part of the data base.

The data is defined on a rectilinear grid with a resolution of 208x278x64 cells with 1200 time steps. The visualized data is a sequence of 2D slices from the three dimensional data set. Each texture has a resolution of 512x512 pixels and uses 40,000 spots. Bent spots were used because of the turbulent nature of the flow in which strong fluctuations for the flow magnitude and direction occur. Each spot is represented as a mesh with a resolution of 16x3 vertices.

Figure 6 is a snapshot of a slice of the data set. The flow is from left to right and impinges a block placed in the field. One can clearly see vortex shedding and the transition from laminar to turbulent flow behind the block. The bottom left image is a detail of the wake behind the block. The resolution of this region of interest is 128x70. The bottom right image is a detail of the detail. The resolution of this region of interest is 22x14. By analysing these details, the user is able to gain insight in the creation of vortices in the wake of the block. In addition, the user can also monitor how the wake behaves over time.

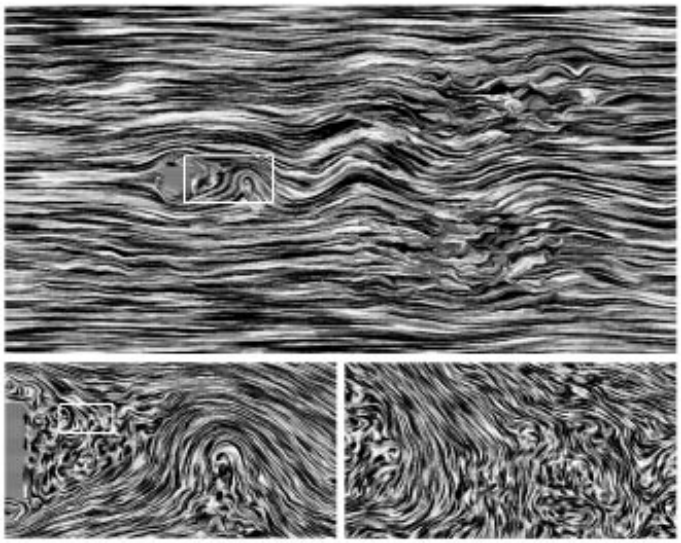

Figure 6: Global and detailed views of a wake behind the block.

4.2 Global climate modeling.
Spot noise has been applied to a number of problems related to weather forecasting and atmospheric pollution. A particular topic of study is the temporal relation between a pollutant and the wind field. The input of these models is often a set of wind, temperature and emission fields. For example, atmospheric pollution researchers are interested in tracking the evolution of smog and how this relates to the wind field. To visualize these temporal relations, we have used time dependent spot noise for the wind field and a color mapping for the scalar field.

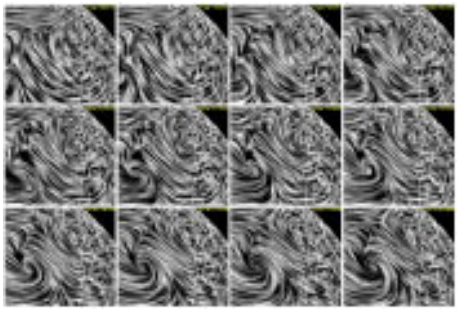

Figure 7: Time series of the global wind field.

Figure 7 is a series of snapshots of the wind field over a three day time period. This series displays a detailed view of the evolution of a depression area.

Figure 8 is a example showing the relation of the temperature and a wind field. In this case, color indicates the temperature and spot noise is used to visualize the wind field. The data originates from global wind and temperature measurements, and was sampled at 12 hour intervals. This application used 62 samples. The data is defined on a 144x72x16 grid. During animation, the user can clearly monitor the dynamic behavior of the temperature, and how this relates to the wind field.

Time steps between two data sets may be too large to get smooth animation of the flow (in this

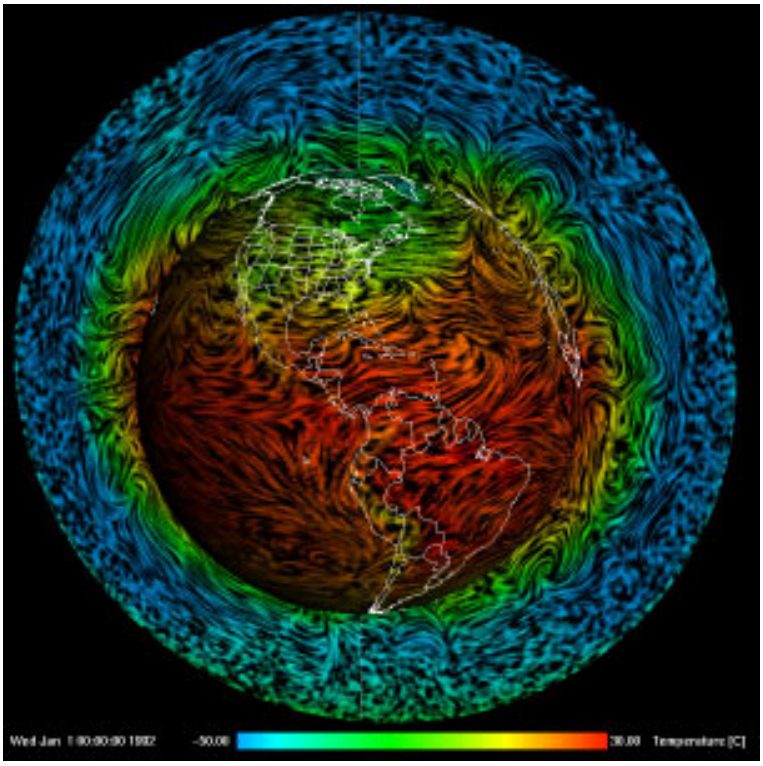

Figure 8: Global view of a temperature and wind field.

application wind fields were sampled at 12 hour intervals). Independence between the time of the data time step and the animation time step is desirable. We have implemented a layer which allows the retrieval of a field value at an arbitrary time. This is achieved by linear interpolation between data time steps. This technique enabled us to produce figure 7, in which the time interval between images is six hours.

5. Conclusion

In this paper, the zooming and time dependent extensions of the spot noise technique have been presented. These extensions allow spot noise to be used for very detailed analysis of time dependent flow fields. The case studies presented in this paper give us confidence that these visualization techniques can now be used to spot structure in complex time dependent flows.

References

1. L. Hesselink and T. Delmarcelle. Visualization of vector and tensor data sets. In L.J. Rosenblum et al., editors, *Scientific Visualization: Advances and Challenges*, pages 419–433. Academic press, 1994.
2. J.J. van Wijk. Spot noise – texture synthesis for data visualization. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 263–272, July 1991.
3. W.C. de Leeuw and J.J. van Wijk. Enhanced spot noise for vector field visualization. In G.M. Nielson and D. Silver, editors, *Proceedings Visualization '95*, pages 233–239. IEEE Computer Society Press, 1995.
4. W.C. de Leeuw and R. van Liere. Divide and conquer spot noise. In *Proceedings Supercompting '97*. ACM, 1997 (to appear).
5. H.W. Shen and D.L. Kao. UFLIC: A line integral convolution algorithm for visualizing unsteady flows. In R. Yagel and H. Hagen, editors, *Proceedings Visualization '97*, pages 317–322. IEEE Computer Society Press, 1997.
6. R.W.C.P. Verstappen and A.E.P. Veldman. Direct numerical simulation at lower costs. *Journal of Mathematical Engineering*, June 1997.