2D Vector Field Visualization Using Furlike Texture

Leila Khouas and Christophe Odet and Denis Friboulet

Creatis, CNRS Research Unit (UMR 5515)
affiliated to INSERM, Lyon, France
khouas@creatis.insa-lyon.fr

Abstract. This paper presents a new technique for 2D vector field visualization. Our approach is based on the use of a furlike texture. For this purpose, we have first developed a texture model that allows two dimensional synthesis of 3D furlike texture. The technique is based on a non stationary two dimensional Autoregressive synthesis (2D AR). The texture generator allows local control of orientation and length of the synthesized texture (the orientation and length of filaments). This texture model is then used to represent 2D vector fields. We can use orientation, length, density and color attributes of our furlike texture to visualize local orientation and magnitude of a 2D vector field. The visual representations produced are satisfying since complete information about local orientation is easily perceived. We will show that the technique can also produce LIC-like texture. In addition, due to the AR formulation, the obtained technique is computationally efficient.

1 Introduction

In many areas such as fluid dynamics, electomagnetics, weather or medical imaging, the analysis of studied phenomena produces complex data that consist of large vector fields. The vectors represent some characteristic of each point on the field such as: a fluid vorticity, a wind velocity or a motion speed. The visualization of vector fields is not straightforward because they have no natural representation. However, many advances have been realized in this research area over the past few years. A state of the art of techniques used in fluid dynamics is given in [1][2]. Basic techniques are iconic representation, particles traces (pathlines) and streamlines, flow topology. More suitable techniques for global flow visualization are those based on the use of texture like the *Spot Noise* [3] and the *LIC* [4] techniques.

Here, we propose to build a new representation of vector fields based on furlike texture. We assume that this will provide a natural representation of a dense vector field. Such an idea has already been mentioned by Kajiya and Kay in [5] but using their fur synthesis technique (based on full 3D texture modeling) would be computationally expensive. Our approach consists of the development of a texture model that allows 2D synthesis of furlike texture having a 3D aspect. The model is based on a non stationary two dimensional Autoregressive synthesis

(2D AR). This provides a simple and efficient 2D generator of furlike texture which is then applied to vector fields representation. The work presented in this paper, enables one to encode in a natural way complete information of a vector field (direction and magnitude) in a still image. Recently, a variant of the *LIC* technique that gives similar results, has been proposed in [6]: the *OLIC* technique. A fast implementation of this technique called *FROLIC*[7] has also been described. We will show that with our completely different approach, we can achieve a computation efficiency that can be compared to the *FROLIC* technique.

Section 2 is devoted to a brief description of the AR based texture modeling [8]. Then, results obtained by applying this model for 2D vector fields visualization are shown in section 3.

2 A 2D texture model for furlike texture synthesis

Several approaches were considered to perform 3D synthesis of realistic furlike texture. Some of the proposed solutions are based on a geometrical modeling of each individual filament. Others attempt to reproduce the complex surface appearance through 3D textures and lighting models [5][9]. These techniques yield realistic results but remain too time consuming. As an alternative to the above complex models, we propose to build a 2D texture model for a simple and efficient synthesis of furlike texture. This requires producing furlike texture with arbitrary orientation and length (i.e. length of individual filaments); in addition, the appearance of the generated texture is desired to be quite realistic. We have used the AR approach described below to build such a model.

2.1 AR modeling of Texture

The AR modeling of a texture is based on a supposed linear dependence of each pixel of the texture image on its neighbors. A texture image is then considered as the output of a 2D linear filter in response to a white noise (Figure 1).

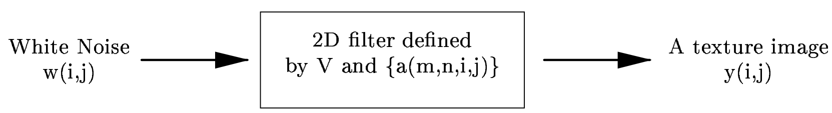


Fig. 1. 2D Autoregressive Filter

Let y be a texture image to be modelized, with a zero mean and $y(i,j)$ the value of a pixel (i,j). The AR modeling of this texture can be described by the following difference equation:

$$y(i,j) = w(i,j) + \sum_{(m,n)\in\mathcal{V}} a(m,n,i,j)y(i-m,j-n) \qquad (1)$$

w is a zero mean white noise with variance σ_w^2. $\mathcal{V}$ defines the prediction region, i.e. the set of neighbors on which the pixel $y(i,j)$ depends (Figure 2). The shape and size of the neighborhood respectively define the causality of the model and its order. $a(m,n,i,j)$ are the model parameters which characterize the texture. In the stationary case, the values of the parameters $a(m,n,i,j)$ are independent of the pixel position (i,j).

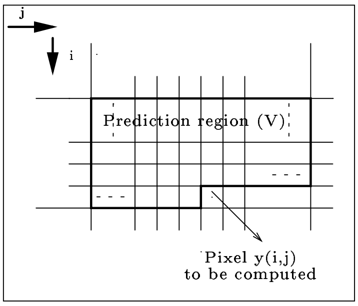


Fig. 2. Prediction region

In our case, we are thus looking for the set of AR parameters that best fits a given stationary furlike texture y. Given an appropriate neighborhood $\mathcal{V}$ (shape and size), the parameters $a(m,n)$ can be estimated using the knowledge of the autocorrelation function (ACF) of the texture image y [10].

Indeed, if we note the ACF by $r(.,.)$, the parameters $a(m,n)$ of the AR model can be obtained by solving the following set of linear equations:

$$r(s,t) - \sum_{(m,n)\in\mathcal{V}} a(m,n)r(s-m,t-n) = 0, \forall(s,t) \in \mathcal{V} \qquad (2)$$

and the input noise variance is computed using:

$$\sigma_w^2 = r(0,0) - \sum_{(m,n)\in\mathcal{V}} a(m,n)r(m,n) \qquad (3)$$

2.2 AR Synthesis of Furlike Texture

In order to build the AR model for furlike texture, it was necessary to have a set of real fur images with different characteristics. Since we needed precise knowledge of length and orientation for each image, we have not used real fur. Instead, we have used as input, images generated by a 3D texture synthesis technique based on the hypertexture modeling developed by K. Perlin and M. Hoffert [9]. This method in which we have introduced orientation, allows a full control of all visual aspects of the texture but, as mentioned earlier, is computationally intensive.

Based on one stationary reference image generated by the oriented hypertexture model and corresponding to one fixed orientation and length, we estimate the ACF $r(m, n)$. Then we compute the AR parameters and the noise variance using equations (2) and (3). The synthesis process consists then of the simple 2D linear filter defined by equation (1) with the corresponding AR parameters.

The neighborhood is chosen so that a causal synthesis can be performed (all pixels in the prediction region are computed before the current pixel). Its shape and size are defined by two parameters jx and jy illustrated in Figure 3. This neighborhood allows a synthesis of orientations in a given range of orientation values (Figure 3). This is due to its causality and its form. We describe in section 2.3 how we deal with any orientation value.

The white noise used as the filter input was chosen to be a 2D Random Sequence of scaled Impulses (RSI) of a fixed magnitude *val* and with an occurrence frequency *p*. Due to its impulsional nature, such a noise allows one to synthesize images having visually well defined individual filaments. Moreover, this noise enables to easily control the apparent filaments density with the parameter *p*.

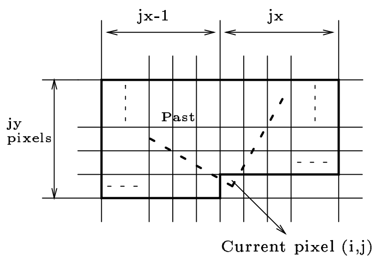


Fig. 3. Neighborhood $\mathcal{V}$

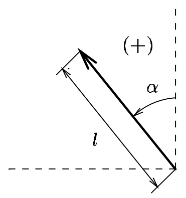


Fig. 4. Coordinates system used to represent the two texture attributes: length l and orientation α

To produce non stationary texture (with varying orientation and length), we use a lookup table of AR parameters, indexed by length and orientation [8] (see Figure 4). This table covers the ranges: $[0, 0.5]$ with a step of 0.05 for length(l) and $[-\frac{\pi}{6}, \frac{\pi}{3}]$ with a step of 1 degree for orientation (α) (see section 2.3).

We present in Figure 5 some results. The first row contains reference images (5.a,5.b) generated with the oriented hypertexture model and images (5.a',5.b') obtained by an AR synthesis with parameters computed from the ACFs of images 5.a and 5.b and using as input noise, a 2D sequence of impulses with parameters val=255 and p=0.033. This noise gives strong directional and depth aspects which are of a great interest for the expected application. The second row contains results of non stationary synthesis: image 5.c is a mosaic of some different orientations; and image 5.d is a furlike texture with constant orientation and length increasing from left to right.

The 2D texture model allows correct synthesis of orientations in some values range. This is due to the causality of the neighborhood and its form (Figure 3).

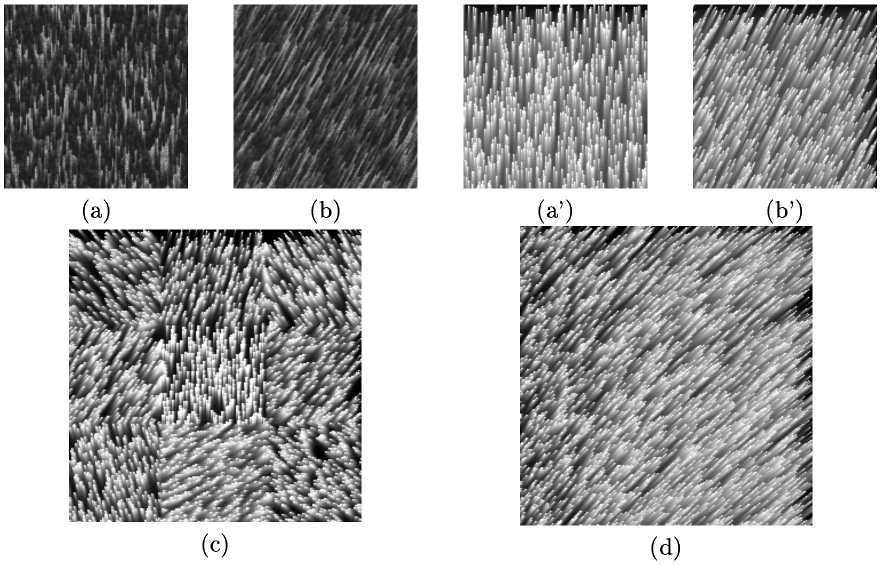


Fig. 5. AR modeling; (a,b) original images. (a',b') corresponding images obtained by AR synthesis using the sizes (jx,jy)=(3,2), (4,2). (c,d) are obtained by non stationary synthesis

2.3 Dealing with any orientation

The model built for furlike texture synthesis allows a correct production of orientations α in a bounded range of values. This range, which is determined by experimentation, is about $\{-\frac{\pi}{6}, \frac{\pi}{3}\}$. This is due to the causality of the neighborhood and its form (Figure 3). To deal with any orientation value, the synthesis procedure consists of four passes through the image with different orders of scanning. Each pass will generate all orientation values fitting in a convenient range as given in Figure 6. To keep a correct synthesis across the boundaries of the passes, the ranges of two successive passes present a small overlapping.

3 2D Vector Field Visualization

3.1 Visualization process

In the previous section, we have built a furlike texture generator allowing a 2D non stationary synthesis of a texture with any orientation value and using length in some available range of values. We will now use this texture model to represent 2D vector fields defined on a 2D support. We assume that the support is a regular grid. A 2D vector given in polar coordinates (α, l) (according to the coordinates system shown in Figure 4) is defined on each point of the grid. The visualization process consists of the synthesis of a furlike texture whose

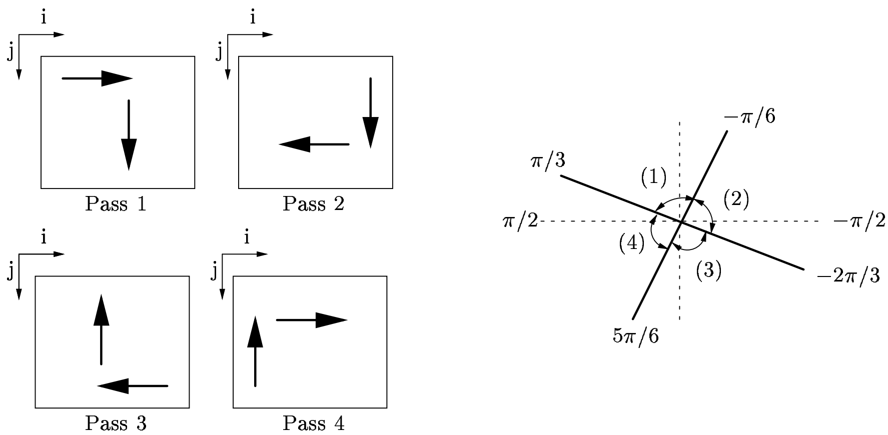


Fig. 6. Orders of scanning the image for the four passes of the synthesis process, with the ranges of orientations produced by each pass

attributes (orientation, length, color, ...) at each point depend on the vector value at the corresponding grid point.

In the following, visualization results obtained using simulated data are presented. We have computed fields similar to those produced around several critical points of different types. Critical points have a zero vector field magnitude. Their study (their location and characteristics) is useful for the analysis of the vector field topology [11][2].

The orientation component (coordinate α) of a vector field is naturally associated to the orientation attribute of the furlike texture. The magnitude component (coordinate l) of the field can be either represented via the length attribute of the texture or encoded using the color or the density.

In all visualizations shown in the following, the RSI noise parameters are set to $p = 0.02$ and $val = 255$.

3.2 Results

Figure 7 presents the result of the visualization process. The critical points are all attracting nodes with different shapes (circle, spiral, ...). The texture length is fixed to $l = 0.20$, the size of the neighborhood is defined by $jx = 5$ and $jy = 2$.

The visual representations obtained seem satisfying since complete information about local orientation is visually perceived. In particular, the depth aspect of the texture makes the direction information non ambiguous even with the dense vector field visualized. This remains true with more complex vector fields such as those presented in Figure 8.

A complete representation of a vector field can be obtained by encoding its magnitude by the length attribute of the texture. For each vector field representation, all the available values of texture length are used in such a way that

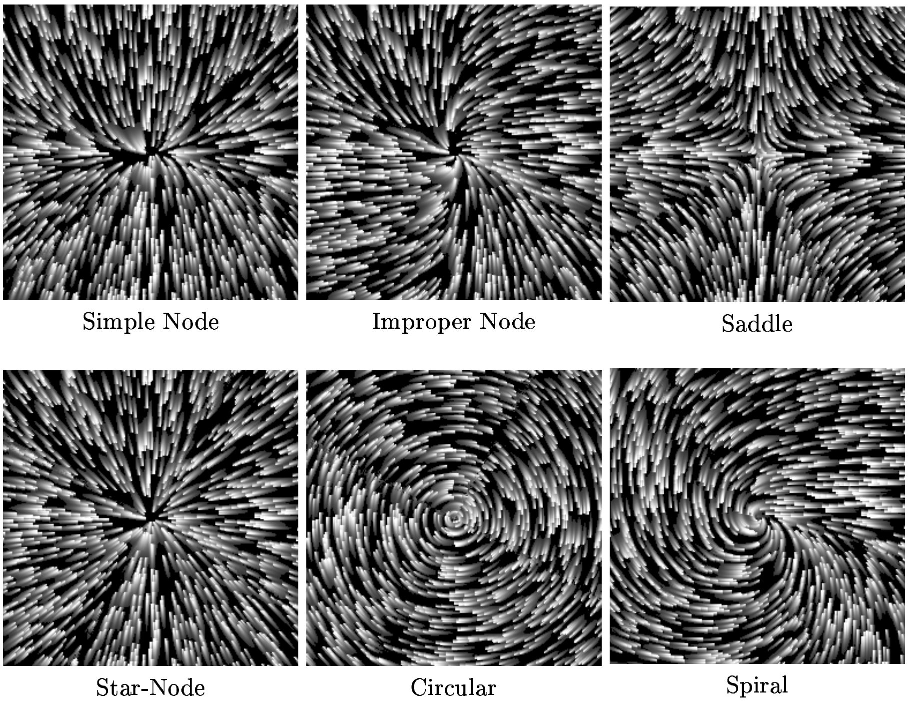


Fig. 7. Visualization of the orientation component of a vector field

greatest values correspond to the regions of highest magnitude. The images obtained are shown in Figure 9. We use the same fields as in Figure 8.

The length attribute gives in a natural way a global information about the spatial variation of the magnitude component. However, since the length is a texture regional attribute, it is difficult to use it to encode a point characteristic (the field magnitude), especially when this latter varies quickly. Note however that a solution could consist of computing the texture at a higher resolution than the original field resolution.

A more precise representation of the magnitude component of a vector field can be achieved by using the color attribute. For this aim, a standard colormap is used in which high values of magnitude are coded by the hottest colors (Red represents the highest value). In Figure 10 (see also the color plate at the end of the book), the same fields as in Figure 8 are visualized using orientation and color attributes of the texture.

Our technique can also produce *LIC*-like texture by using a fine white noise instead of the impulsional one used to generate furlike texture. In Figure 11, we show the result of visualizing the same fields as in Figure 8 by simply using a gaussian white noise as input of the AR filter.

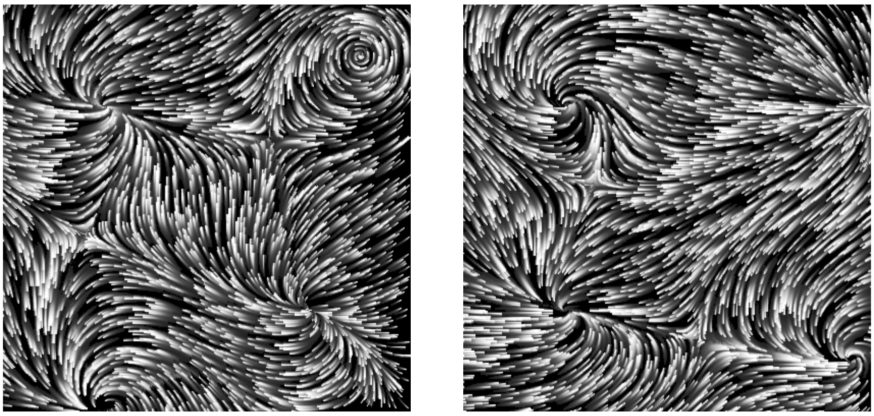


Fig. 8. Fields with several critical points ($p = 0.02$ and $l = 0.5$)

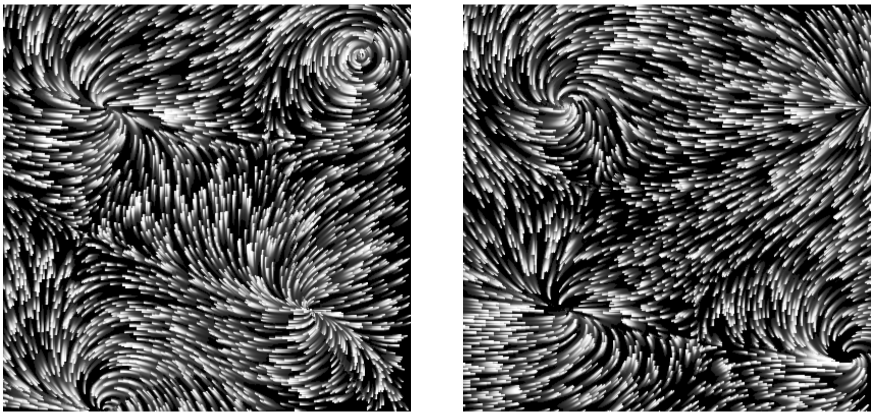


Fig. 9. Length attribute to encode the magnitude ($p = 0.02$)

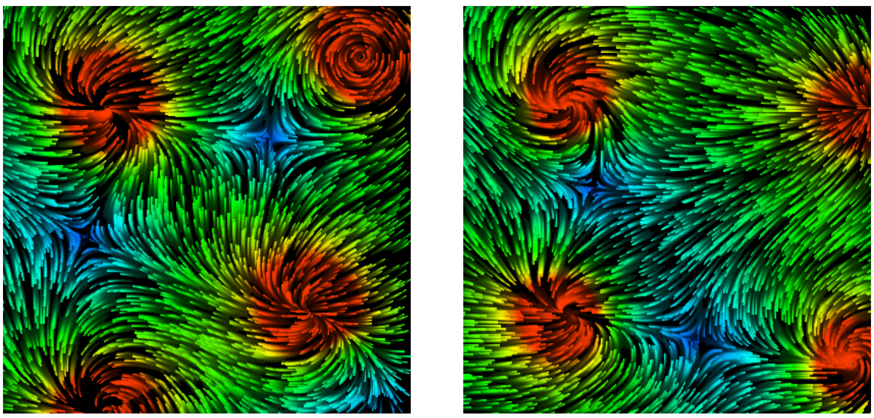


Fig. 10. Color encoding for the magnitude ($p = 0.02$ and $l = 0.5$)

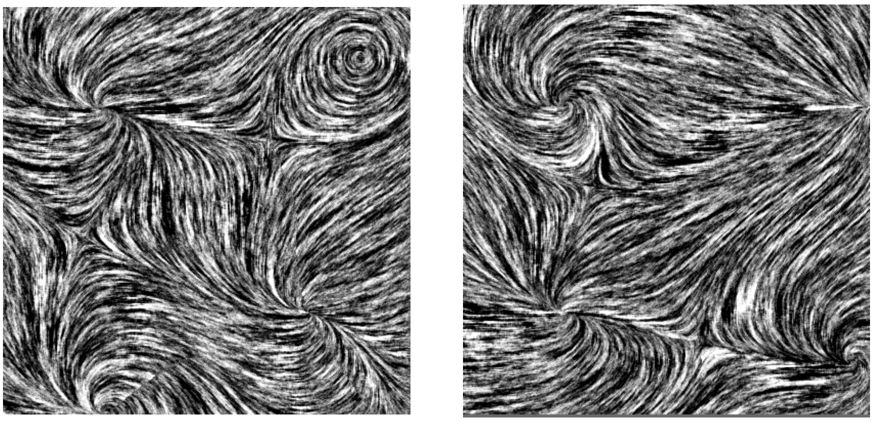


Fig. 11. *LIC*-like images produced by the AR filter with a gaussian white noise ($l = 0.5$)

3.3 Implementation and Complexity

Given a 2D vector field defined on a grid with N data points, the visualization process complexity can be described as follows:

- A noise image (N pixels) (RSI) with the desired density is generated.
- Four passes via the AR filter are performed (the output of one pass is used as the input of the next one). Each pass performs the following operations for each pixel (i, j) (among the N):
  - The vector field value at the corresponding data point is retrieved. It consists in two values $(\alpha,\ l)$. If the orientation do not belong to the range values of the current pass, the pixel is skipped. Else, α and l are used as input indices to the lookup table of AR coefficients. A set of $M = 2jxjy - jx - jy$ coefficients $a(m, n)$ is then returned. M is the size of the neighborhood which is defined by the two dimensions jx and jy.
  - the value at pixel (i, j) is computed using the filter difference equation (1). The computation consists in M additions and M multiplications.
- Thus, the four passes involve near $2N(2jxjy-jx-jy)$ arithmetic operations. This number depends only on the texture resolution and the neighborhood size (which is fixed to the convenient values $jx = 5$ and $jy = 2$).

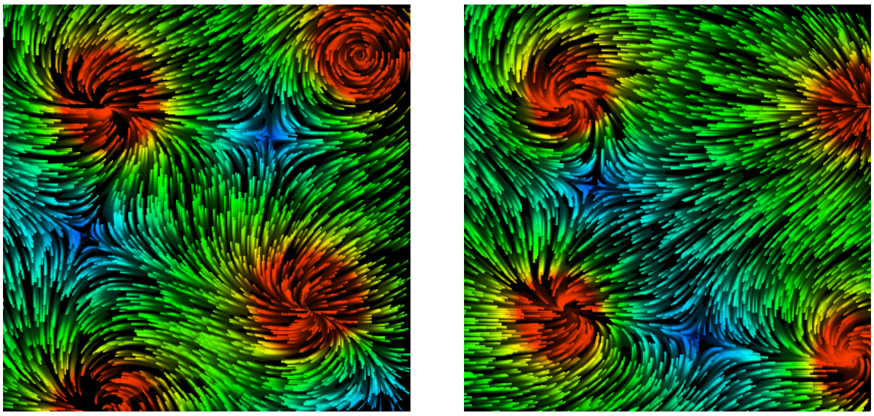
The technique is implemented in C language on an SGI station with MIPS R4400 200Mhz processor. As an example, each one of the images presented in Figure 10 is produced in less than 1 *second* (this time includes all the steps of the visualization: noise generation, four passes). The resolution of the images is 400×400. Thus, our technique allows one to achieve a time computation efficiency comparable to the most recently proposed techniques such as the *OLIC*[6] and *FROLIC*[7] which give similar results.

4 Conclusion

In this work, we have presented a new technique for vector fields visualization based on the use of a furlike texture. For an efficient generation of such texture, we have proposed a 2D texture model based on a 2D AR synthesis. We have shown that the use of an AR model gives satisfying visual results, even with very small sizes of neighborhood. This makes the technique fast. In addition, several texture attributes can be locally controlled (orientation, length, density via the noise parameter, color). We have demonstrated the use of this texture model for the visualization of 2D vector fields. We have shown that several representations can be easily generated. In particular, *LIC*-like representations can also be produced. As future work, we expect to animate the visualization and apply the technique developed to real data and to unsteady flow. Another perspective consists of the application of the technique to visualize 3D vector fields on 3D surfaces by introducing appropriate texture mapping procedures.

References

1. V. Watson and P. P. Walatka, "Visual analysis of fluid dynamics," in *State of the Art in Computer Graphics - Visualization and Modeling* (R. David and E. Rao, eds.), pp. 7–18, New York: Springer, 1991.
2. F. H. Post and J. J. Van Wijk, "Visual representation of vector fields: recent developments and research directions," in *Scientific Visualization, Advances and Challenge*, pp. 368–390, San Diego: Academic Press, 1994.
3. J. J. Van Wijk, "Spot noise," *Computer Graphics*, vol. 25, pp. 309–318, July 1991.
4. B. Cabral and L. Leedom, "Imaging vector fields using line integral convolution," *Computer Graphics*, vol. 27, no. Annual Conference Series, pp. 263–270, 1993.
5. J. T. Kajiya and T. L. Kay, "Rendering fur with three dimensional textures," *Computer Graphics*, vol. 23, pp. 271–280, July 1989.
6. R. Wegenkittl and E. Groller, "Animating flowfields: rendering of oriented line integral convolution," in *Proceedings of Computer Animation'97, Geneva, Switzerland, June 5-6, 1997*, pp. 15–21, June 1997.
7. R. Wegenkittl and E. Groller, "Fast oriented line integral convolution for vector field visualization via the internet," in *Proceedings of Visualization '97, Phoenix, Arizona, Oct 19-24, 1997*, pp. 309–315, Oct. 1997.
8. L. Khouas, C. Odet, and D. Friboulet, "3D furlike texture generation by a 2D autoregressive synthesis," in *The sixth International Conference in Central Europe on Computer Graphics and Visualisation*, pp. 171–178, Feb. 1998.
9. K. Perlin and E. M. Hoffert, "Hypertexture," *Computer Graphics*, vol. 23, pp. 253–262, July 1989.
10. S. M. Kay, *Modern spectral estimation: theory and application*, pp. 479–518. Prentice Hall, 1988.
11. A. R. Rao and R. C. Jain, "Computerized flow field analysis: oriented texture fields," *IEEE Trans. on Pattern Anal. and Machine Intel.*, vol. 14, pp. 693–709, July 1992.



2D Vector field visualization using an autoregressive synthesis of
furlike texture with color encoding of the magnitude