Image-based Reconstruction and Interaction of 2D Vector Fields

Carlos Pérez Risquet [†] and Holger Theisel [‡]

[†] Central University of Las Villas, Computer Science Department
Carr. a Camajuaní. Km. 5½, Santa Clara, Cuba
Email: cperez@uclv.edu.cu

[‡] University of Rostock, Computer Science Department, Institute for Computer Graphics
A.-Einstein-Str. 21, 18051 Rostock, Germany
Email: theisel@informatik.uni-rostock.de

Abstract

This paper addresses the problem of symmetrical 2D flow visualization and presents solutions to compute a vector field from a flow image. Based on these solutions new, interactions mechanisms are developed which make visual corrections of 2D vector fields possible. These mechanisms permit an image-based interaction with some features of the vector field: vectors, streamlines, and critical points.

1 Introduction

The visualization of flow data has become an important and well-developed part of scientific visualization. Vector fields to be visually analyzed appear in a variety of applications such as fluid mechanics, climate modeling, dynamical systems, electromagnetism, and material engineering. A variety of techniques for visualizing and analyzing vector fields have been developed.

Interaction is a process inherent to visualization. Large data sets can only efficiently be analyzed by the use of the interactive exploration coupled with the appropriate presentation mechanisms. Due to the high amount of present data, this statement is particularly true for flow visualization. A problem to be solved here is the access to the flow data from the images, that is, through the identification of points in the image. In general this is only possible if the corresponding information is coupled to these points. One general solution to this problem is presented in [4], where the concept of symmetric input/output in the visualization process is considered.

It is the purpose of this paper to study a symmetric visualization for the particular case of 2D flow data. The classical visualization pipeline for flow visualization transforms flow data to a visual representation. In order to establish a symmetric visualization, we particularly have to deal with the inverse problem: how to reconstruct the underlying original flow data from a given visual representation. This information should be obtained only from the data contained in the image. Based on solutions to this problem we can then introduce new image-based interactions methods for vector fields: given the visualization of a vector field, the user/designer is allowed to modify certain features in the visualization while the underlying vector field is automatically corrected.

Therefore, for an image-based interaction of vector fields, the following steps are necessary:

1. a global visualization (image) of the vector field.
2. algorithms to calculate a vector field from an image.
3. suitable interaction mechanisms to modify the flow image.

The rest of the paper is organized as follows. Section 2 analyzes and selects appropriate visualization techniques for which a reconstruction of the original vector field is possible. Section 3 introduces methods to reconstruct vector fields from their visualizations. Based on these results, new image-based interaction techniques for the transformation of 2D vector fields are introduced in section 4. Finally, section 5 shows the conclusions of this work and outlines the future work.

2 Selection of an appropriate visualization technique

In this section we analyze which flow visualization techniques are appropriate for the accomplishment of a symmetrical visualization process. Detailed descriptions of flow visualization methods can be found, for example, in [3], [5] and [12].

Approaches for flow visualization can be classified in three groups: elementary, local and global methods [3]. Elementary and local methods show properties of the field only at selected points of the flow and in the vicinity of these points. Therefore, they produce images, which in many points do not contain information about the vector field. Because of this, elementary and local methods are not suitable for the realization of symmetrical visualization processes.

Global methods for flow visualization code information of the vector field at each image point. Texture-based methods and the analysis and representation of the vector field topology belong to this group. Another approach for the construction of a global visualization is based on reducing the vector data to scalars and to visually code these values.

Among the most popular and well-established texture-based visualization methods are *Spot Noise* [17] and *Line Integral Convolution* (LIC) [2]. These methods have been extended and improved in many further publications [1], [8], [9], [13], [14], [15], [18].

In general, texture based methods rely on distorting a given texture according to the properties of the flow field. Figure 1a) shows a LIC-image of the electric field of a dipole antenna. Due to the nature of the algorithm, the resulting images tend to be fuzzy and locally blurred. In fact, each image point shows information about the flow direction, but this information is too noisy to be automatically extracted from the image. Figure 1b) illustrates a zoomed region of figure 1a). The trace of individual streamlines cannot be exactly distinguished, because these are blurred and they merge into the image. An exact determination of the streamlines direction is not possible in this case.

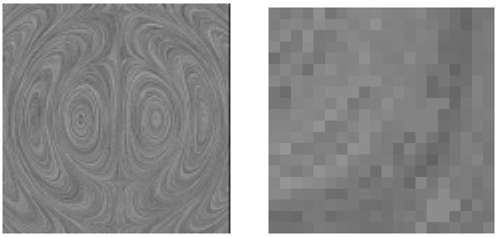
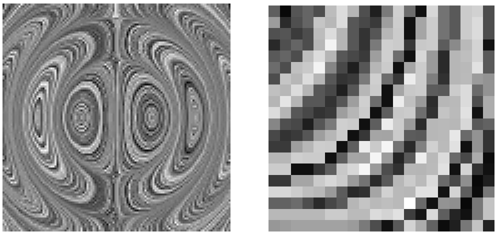

a) b)

Figure 1. a) LIC-image of the electrical field of a dipole antenna, b) zoomed region of a). (Data set property of Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany)

A global method for the visualization of 2D vector fields that produces flow images with a clearer appearance of particular streamlines is presented in [10]. This technique, called *Integrate and Draw*, was especially developed to make the computation of vector data from flow images possible. The main idea of the method consists of drawing each streamline with a (random) different color. If two or more streamlines coincide at a point, the average of the gray levels of these streamlines is calculated in this point. Figure 2a) shows the data set of the dipole antenna visualized with *Integrate and Draw*. The improved definition of the streamlines can be confirmed here, and the individual trace of each one can better be distinguished. Figure 2b) shows the same zoomed region of the vector field as in figure 1b) using *Integrate and Draw*.

Due to the advantages of this method, the following studies on interaction with the flow images are based on *Integrate and Draw*.



a) b)

Figure 2. a) *Integrate and Draw*-image of the electrical field of the dipole antenna, b) zoomed region of a) (same region as in figure 1b))

3 Computation of a vector field from a flow image

The main idea to calculate a vector field from a flow image depicting the streamlines is to determine the direction of the streamline that passes through each image point. Then the original vector field is constructed in the following way:

- The vector field is constructed on a regular mesh, which dimensions correspond to the dimensions of the image.
- The direction of the vector at each mesh point is calculated by determining the direction of the streamline at the corresponding image point.
- Since no information about the magnitude of the vectors is encoded in the image, the length of all vectors is set to 1.

We applied three algorithms were applied for the determination of the vectors associated with each image point: the search-rays technique, principal axes, and the best paths method.

The search-rays and principal axis techniques are used in image processing for edge detection, feature extraction and object recognition. These methods were modified in order to calculate the direction of a streamline in a flow image. A detailed description of this can be found in [11].

The *best paths* method for the recognition and tracking of streamlines in flow images will be briefly described now.

First of all we need a few definitions. Let I be a digital image, and $x_0, x_1, \ldots, x_n$ points in I.

Definition 1 (Consecutive neighbors). The points $x_0, x_1, \ldots, x_n$ are called *consecutive neighbors,* if x_i is a neighbor of x_{i-1} for $i = 1, 2, \ldots, n$. The concept of neighborhood used here is as usual in image processing.

Definition 2 (Path). A series of consecutive neighboring points $x_0, x_1, \ldots, x_n$ with initial point x_0 and length $n+1$ is called a *path of length n starting from* x_0, and is represented by $W_n(x_0)$.

Let M be the set of all points of all paths of length n starting from x_0, and let C_n be a function of I^{n+1} in $\mathbb{R}$,

$$C_n: I^{n+1} \text{ ? } \mathbb{R},\ (x_0,\ldots,x_n) \text{ ? } C_n(x_0,\ldots,x_n).$$

We consider the restriction of C_n to M, $C_n|M$. Since the image is digital, the number of paths is finite. This implies that $|M|$ is also finite and the restricted function $C_n|M$ reaches its maximum and its minimum in M. We are interested in the paths $W_n(x_0)$ where the function $C_n|M$ reaches its minimum.

Definition 3 (Best paths). The paths where the function $C_n|M$ reaches its minimum are called *best paths of length n starting from* x_0 *relative to* C_n, and they are represented by $W_n^*(x_0)$[1].

In summary it can be said that a *best path of length n starting from* x_0 *relative to* C_n *is a series of consecutive neighboring points* $x_0,\ldots,x_n$ so that the function C_n reaches its minimum at $(x_0,\ldots,x_n)$.

These concepts can be used for the solution of our problem. If we define a function C_n^* in such that way that it reaches its minimum over the points that belong to the streamline, and only over them, then the streamlines recognition problem is reduced to an optimization problem for which sufficiently efficient algorithms exist. The streamline can then

---

[1] In general there is not a unique best path.

be expressed as a best path relative to the function C_n^*.

The next question is, how to define C_n^*? Some general properties this function should satisfy are:

- *Find the streamlines*: By minimizing C_n^* one should obtain the points $x_0,...,x_n$ that form the streamlines.
- *Minimum curvature*: The line connecting the points $x_0,...,x_n$ should have minimum curvature. This condition avoids cycles in the path found.
- *Incremental*: This property is not strictly necessary but improves the minimization of C_n^*. Incremental means that $C_n^*(x_0,...,x_n)$ only depends on x_n and $C_{n-1}^*(x_0,...,x_{n-1})$.

A candidate of such a function is,

$$C_n^*(x_0, x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} k(x_i)\left|g(x_i) - \overline{x}_{i-1}\right|$$

$$= \frac{n-1}{n} C_{n-1}^*(x_0, x_1, \ldots, x_{n-1}) + \frac{k(x_n)}{n}\left|g(x_n) - \overline{x}_{n-1}\right|$$

where:

- *$g(x_i)$ is the gray level of x_i,*
- *$\overline{x}_i$ is the average of the gray levels of $x_0,...,x_i$, and*
- *$k(x_i)$ is a penalty factor for the curvature between x_i, x_{i-1} y x_{i-2}.*

The usefulness of this function for the recognition of streamlines was verified in [11]. by numerous vector fields. The best paths relative to this function correspond to the streamlines of the flow images. Figure 3 visualizes with the *Integrate and Draw* technique a few examples of original vector fields (upper row) and the corresponding recomputed vector fields from flow images (lower row).

A more detailed analysis of the differences between original and recomputed vector fields by means of distance functions and visual comparisons can be found in [11].

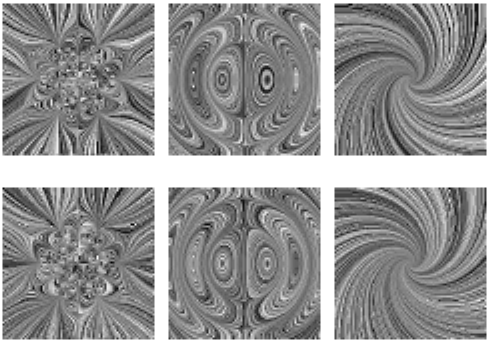


Figure 3. Original (upper row) and recomputed vector fields from *Integrate and Draw* images (lower row)

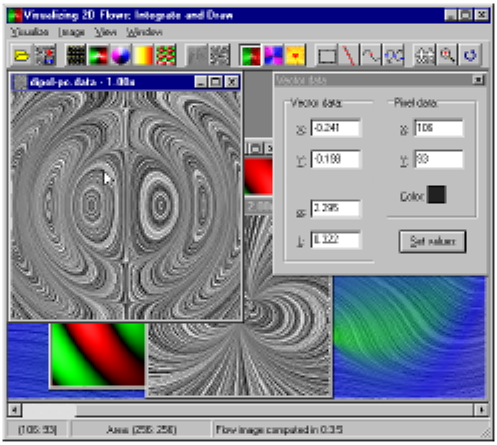
4 Interaction techniques for the transformation of 2D vector fields

This section presents interaction techniques for the visualization of 2D flows. These techniques assume the existence of a global visualization of the vector field (2D image) and an algorithm that calculates a vector field from this image like the ones described in sections 2 and 3. The presented interaction techniques are, however, independent of the algorithm for computing the vector field and the visualization method that is used.

4.1 Interaction with vectors

The interaction with the vectors is a simple but powerful form of interaction in the flow visualization. It can only be accomplished in symmetrical visualization processes, because it presupposes that the user interacts with the vector data that correspond to the pixels of the image. This means, it is possible here to consult and to modify the vector data that correspond to each image pixel. Each modification of the vector field must be reflected immediately in the visualized image.

Figure 4 shows an example of an interaction with the vectors of the data set of the dipole antenna. The visualization of the data fields and the interaction are realized using the system *Flow Studio*, which was developed to verify the concepts outlined here. In the window on the right-hand side the (cartesian and polar) coordinates of the vector corresponding to the cursor position are shown. These data can be modified by writing new values in the corresponding fields. The corresponding vector receives the new values automatically, and the image is updated accordingly.



Vector data: x: -0.241; y: -0.188; 1.295; 0.322

Pixel data: x: 106; y: 93

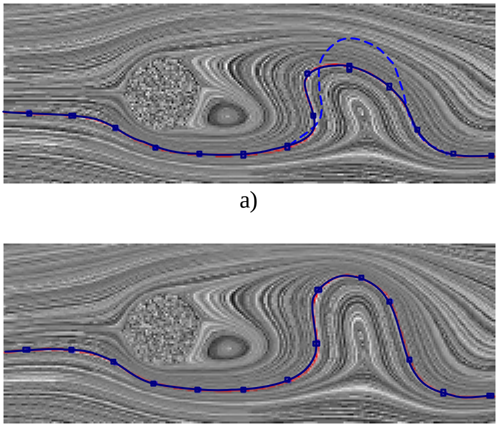
(106; 93) | Area (256; 256) | Flow image computed in 0.35

Figure 4. Interaction with the vectors in a symmetrical visualization process

4.2 Interaction with the streamlines

The interaction with the streamlines was one of the strongest motivations for the development of symmetrical visualization processes. In this case the user is going to interact with the flow data by means of streamlines. By clicking on the image not only the pixel under the cursor is selected, but also the complete streamline that passes through that point. This requires not only the computation of the vector data that corresponds to the current cursor position, but of all the vectors that belong to the streamline.

By interactively modifying a streamline a user is able to modify the flow in certain regions of the vector field. By modification of streamlines we understand dragging and deformation of streamlines.



a)

b)

Figure 5. Interaction with the streamlines in a symmetrical visualization. a) the selected streamline is interactively modified, b) results of the modification

Figure 5 shows an example of the deformation of a streamline. The continuous line in a) was identified and calculated according to the mechanisms explained in section 3. This line should be transformed such that it follows the route of the discontinuous line. For better visibility only the part of the discontinuous line that do not overlap the continuous one is shown.

The transformation of a flow image by dragging and/or deforming the streamlines can be achieved by the application of warping techniques. In [6] different methods for that purpose are presented from which the mesh-based and the feature-based techniques are most commonly used.

Obviously, the change of a streamline in a flow image causes the change of many other streamlines and possibly the change of the entire picture. Even if only local transformations in a certain area are desired, with many warping techniques the preservation of the remaining picture is not completely ensured.

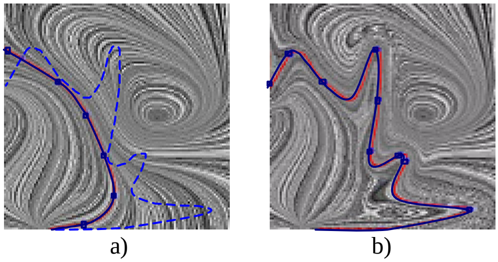


Figure 6. Restrictions of the interaction with the streamlines in a symmetrical flow visualization. a) visualization of a simple vector field in which the continuous streamline is to be changed. b) result of the change

The transformation of the streamlines of a flow image in a symmetrical flow visualization has certain restrictions and should therefore be used with care. That is, the streamlines are not arbitrary to be changed. Small changes in the geometry of the streamlines can be made, e.g., in order to correct fine details in the vector field or to eliminate small simulation errors. Larger changes in the picture should be avoided, however, because the streamlines can be destroyed and the calculation of a vector field is no longer possible. Figure 6 shows an example in which a streamline is changed in such a way that in the resulting image b) new critical points appear, and in some areas the streamlines cannot be recognized any more. In this example, the topology of the vector field and the streamlines was destroyed by the warping process. Thus the process is no longer controllable, and the interaction may have unforeseeable effects.

To guarantee only permitted changes of the vector field, a correct choice of a warping technique and its parameters has to be made. A wrong warping technique can possibly lead to unwanted results. Figure 7a) shows a streamline which was transformed with two different warping techniques b) and c). In c), the result is obviously wrong. Here the *inverse mapping* method with the *field-based* technique for reconstruction of the objects was used. In b), the *2-pass spline mesh* method with the *radial field-based* technique for reconstruction of the objects was applied.

It is not the purpose of this paper to examine in detail the area of morphing and warping techniques. A detailed description of the above specified warping techniques can be found e.g. in [6]. For the purpos of the symmetrical flow visualization it is sufficient to choose and use a suitable technique. We found the *2-pass spline mesh* technique as the best suitable method to transform the streamlines in flow images.

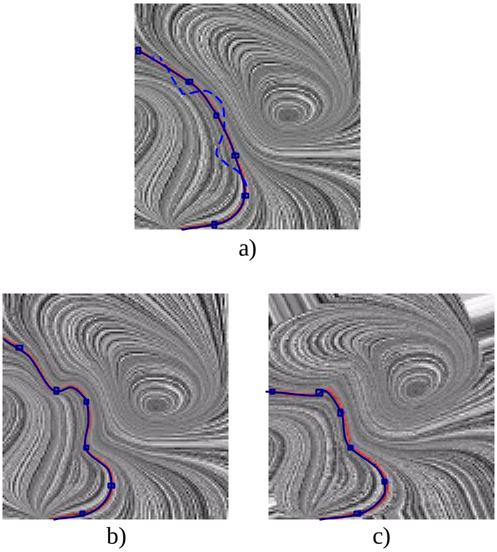


Figure 7. Use of different warping techniques to deform a streamline. a) the highlighted (continuous) streamline is to be transformed into the discontinuous one, b) result of the warping process with the *2-pass spline mesh* technique, c) result with the *inverse mapping* technique

The *2-pass spline mesh* technique is an efficient method that was developed especially for digital pictures (see [19]). Although it is an efficient technology, it demands many interactions and a lot of information from the user in order to define the deformations, since it is based on a net specification. In order to reduce the input of the user, the visualization system could construct and distort the net (semi) automatically. This is especially possible in the symmetrical flow visualization. Since the system knows about streamlines, it can produce a net semi automatically which adapts to the path of the streamlines. Figure 8 illustrates such a net. The horizontal lines are produced automatically as the user clicks on any point of the selected streamline. The vertical net curves must be defined and deformed by the user. For better illustration not all vertical splines are shown.

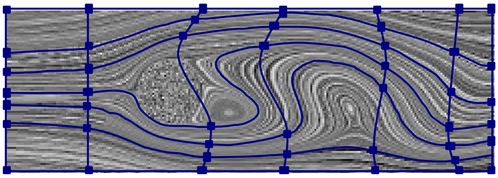


Figure 8. Splines net to transform a flow image

We have to remark that in general not all streamlines of any vector field can be used as pattern for the splines. This is the case, e.g., if the streamline builds a closed curve. However, the splines net can always be refined until all desired details are reached.

Finally it can be stated that certain limits are set to the interaction with streamlines. By local transformations, however, it offers an efficient method to correct of a flow field.

4.3 Topological Interaction

The topology has been shown to be one of the most distinctive features of a vector field [7]. Hence, the interactive change of the topology of a vector field is an interesting candidate for interaction techniques. The topology of a vector field basically consists of critical points and separatrices [7]. Topological interaction in our case means that the user may interactively insert/remove/modify critical points and separatrices in the image. Then the underlying vector field is modified automatically according to these changes.

One approach for the realization of a topological interaction is presented in [16]. There the user can construct a topological skeleton of arbitrary complexity while the system automatically creates a piecewise linear vector field of exactly the specified topology. Using this approach, a topological interaction can be done in the following way:

1. Extract the topology of the vector field.
2. Allow interactive changes of the topological skeleton obtained in 1.
3. Reconstruct a vector field with the topology specified in 2. following [16].

Figure 9 shows an example how a topological skeleton is constructed, and how a corresponding vector field is obtained.

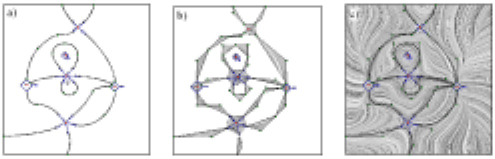


Figure 9. Topological interaction in flow visualization. a) constructing a topological skeleton; b) constructing a piecewise linear vector field for critical points and separatrices; c) piecewise linear interpolation of the remaining parts.

5 Conclusions and future work

The symmetry problem in 2D flow visualization was addressed in this paper. A general approach to compute a vector field from a flow image depicting the streamlines was presented. This approach is based on determining the direction of the streamline that passes through each image point. For this purpose a new technique, called *best paths*, was developed in this work, and two more algorithms of image processing were applied. Based on these solutions, new interaction mechanisms have been developed which allow the visual transformation of 2D vector fields. These mechanisms permit an image-based interaction with features of the vector field.

However, a number of problems are still unsolved considering both the reconstruction of the underlying vector fields from the image and the realization of the interaction. In particular, the following issues are to be addressed in future research:

- The reconstruction of the vector field from the image works rather reliable in regions of homogeneous flow behavior while the results are less reliable in the neighborhood of critical points.

In particular, it cannot be guaranteed yet that the original and the reconstructed vector field have the same topology.

- There are no exact conditions yet which interactions of streamlines are permitted concerning the property of preserving the topology of the original vector field.

Acknowledgements

The authors would like to thank Prof. Heidrun Schumann from Rostock University for her constant encouragement and support of the research presented in this paper.

References

[1] Battke, H.; Stalling, D.; Hege, H. C.: *"Fast Line Integral Convolution for Arbitrary Surfaces in 3D"*, in Hege, H. C. and Polthier, K. (Eds.), Visualization and Mathematics, pp. 181—195, 1997

[2] Cabral, B.; Leedom L.: *"Imaging vector fields using Line Integral Convolution"*, Proc. of SIGGRAPH '93, in Computer Graphics 27, ACM SIGGRAPH, pp. 263–272, 1993

[3] Delmarcelle, T.; Hesselink, L.: *"A Unified Framework for Flow Visualization"*, in *"Computer Visualization. Graphics techniques for scientific and engineering analysis"*, CRC Press, 1995

[4] Felger, W.: *"Innovative Interaktionstechniken in der Visualisierung"*, Springer Verlag, 1995 (in German)

[5] Frühauf, T.: *"Graphisch-Interaktive Strömungsvisualisierung"*, Springer Verlag, 1997 (in German)

[6] Gomes, J.; Darsa, L.; Costa, B.; Velho, L.: *"Warping and Morphing of Graphical Objects"*, Morgan Kaufmann Publishers, Inc., 1999

[7] Helman, J; Hesselink, L.: *"Visualization of vector field topology in fluid flows"*. IEEE Computer Graphics & Applications, Vol. 11, Nr. 3, pp. 36–46, 1991

[8] Interrante, V.; Grosch, C.: *"Visualizing 3D flow. IEEE Computer Graphics and Applications"*, pp. 49–53, 1998

[9] Okada, A.; Lane, D.: *"Enhanced line integral convolution with flow feature detection"*, SPIE Vol. 3017 Visual Data Exploration and Analysis IV, pp. 206–217, 1997

[10] Pérez Risquet, C.: *"Visualizing 2D Flows: Integrate and Draw"*, Proc. of the 9th Eurographics Workshop on Visualization in Scientific Computing, Germany, 1998

[11] Pérez Risquet, C.: *"Visualisierung und Rückrechnung von zwei-dimensionalen Vektordaten"*, Ph.D. Thesis, University of Rostock, Computer Science Department, 2002 (in German)

[12] Post, F. H.; van Wijk, J. J.: *"Visual representation of vector fields: recent developments and research directions"*, in *"Scientific Visualization. Advances and Challenges"*, Academic Press, 1994

[13] Shen, H.W.; Kao, D.L.: *"UFLIC: a line integral convolution algorithm for visualizing unsteady flows"*, Proc. of Visualization '97, pp. 317–322, IEEE, 1997

[14] Shen, H.W.; Kao, D.L.: *"A new line integral convolution algorithm for visualizing time-varying flow fields"*, IEEE Transactions on Visualization and Computer Graphics, pp. 98–108, 1998

[15] Stalling, D., Hege, H. C.: *"Fast and Resolution Independent Line Integral Convolution"*, Computer Graphics Proceedings '95, ACM SIGGRAPH, pp. 249–256, 1995

[16] Theisel, H.: *"Designing 2D vector fields of arbitrary topology"*. Computer Graphics Forum, 21(3), 2002. (Proceedings Eurographics 2002).

[17] van Wijk, J. J.: *"Spot noise texture synthesis for data visualization"*, Proc. of SIGGRAPH '91, in Computer Graphics 25, pp. 309–318, 1991

[18] Wegenkittl, R.; Gröller, E.; Purgathofer, W.: *"Animating Flowfields: Rendering of Oriented Line Integral Convolution"*, in Computer Animation '97, pp. 15–21, IEEE Computer Society Press, 1997

[19] Wolberg, G.: "*Digital Image Warping*", IEEE Computer Society Press, 1990