Function Based 2D Flow Animation

Ergun Akleman, Sajan Skaria, Jeff S. Haberl

Abstract— **This paper summarizes a function-based approach to create 2D flow animations. We use periodic functions to create cyclical animations that represent 2D flow. These periodic functions are constructed with an extremely simple algorithm from a set of oriented lines. The speed and orientation of the flow are described directly by the orientation and the lengths of these oriented lines. The resulting cyclical animations are simply obtained by sampling the constructed periodic functions. Two example applications of 2D flow animations is provided where they are used to improve the visualization of compressor operation.**

Keywords: Visualization, Image Synthesis, Computer Animation, Flow Animation

I. Introduction

In this paper, our goal is to create user-controlled cyclical and variable speed flow animations. Our research in implicit surface modeling provided the theoretical framework for introducing a new approach for animating 2D flows. It is extremely simple to develop a system based on the new approach. The interface to define the flow is also very user-friendly. Based on this approach we have implemented a system and created various 2D flows. We have also used the flow animations to improve the quality of compressor visualizations as part of a research project to investigate how multimedia can improve engineering handbooks (see acknowledgments).

A similar problem, the creation of a 2D flow animation for a given vector field, has been popular in recent Computer Graphics publications. The most common solution to the problem is to use the line integral convolution (LIC) technique proposed by Cabral and Leedom [4]. Forsell and Cohen have also accomplished this task with improved periodic filters which are originally suggested by Freeman, Adelson and Heeger [7]. They modulated the rate of the function phase shift as a function of the vector magnitude to obtain the variable-speed flow animations [5], [6]. Stalling and Hege also improved this technique by eliminating aliasing effects [13].

Max and Becker proposed a texture advancement technique to create cyclical flow images [10]. Van Gelder and Wilhelms used color tables to animate 3D flow fields [8]. Jobard and Leber proposed the idea of motion maps to speed up the computation of steady flow animations [9].

In our work, we do not use a given vector field to create flow animations. Users themselves design the flow by drawing a set of oriented lines. The speed and orientation of the flow are then described directly by the orientation and the lengths of these oriented lines. Unlike previous research, we use an implicit-based approach instead of a line integral convolution (LIC) approach. Our solution also provides a simple approach to develop an intuitive user interface to design flow animations.

E. Akleman is an Assistant Professor in Visualization Sciences Program, Department of Architecture, Texas A&M University. S. Skaria is a graduate student in the same program. Their address: Visualization Laboratory, 216 Langford Center, College Station, Texas 77843-3137. Emails: ergun@viz.tamu.edu, sajan@viz.tamu.edu. J. Haberl is an Associate Professor in Department of Architecture, Texas A&M University. His address: Energy Systems Laboratory, Department of Architecture, Texas A&M University, College Station, Texas 77843-3581. Email: jhaberl@esl.tamu.edu.

II. Methodology

Our approach is based on functions that are constructed by the operations that are used in an implicit surface construction. The flow is described by a set of oriented lines. For each oriented line segment a periodic function is created to describe a flow in the direction and position of the line. Then, these periodic functions are combined using an approximate union operation. The following subsection describes how the periodic function is created for a given line segment.

A. Flow Described by One Line Segment

Let an oriented line be given by two end points v_1 and v_2. Our first goal is to construct a periodic function that will give a flow in the direction of $v_2 - v_1$ with the speed $|v_2 - v_1|$. This function can simply be written as

$$p(v, t; \omega) = \sin\left(t + \omega \, \frac{(v - v_1) \bullet (v_2 - v_1)}{|v_2 - v_1|^2}\right),$$

where ω is the frequency, t is the phase and $\bullet$ scalar multiplication operator. Figure 1 demonstrates how this function can be used to attach a color range to the scale from -1 to 1. By changing t between 0 and 2π in equal intervals, we obtain a cyclic animation that gives an illusion of a flow in the direction of $v_2 - v_1$ with the speed $|v_2 - v_1|$.

Unfortunately, the periodic function $p(v, t; \omega)$ does not indicate the position of the line (Any line with the same direction and length create similar images). Therefore, in order to visually emphasize the line that creates the motion we use a function $d(v)$ that gives minimum distance [3] to the given line segment. The function $d(v)$ is given by the equality

$$d(v) = \begin{cases} |v - v_1| & \text{if } (v_1 - v_2) \bullet (v - v_1) \geq 0, \\ |v - v_2| & \text{if } (v_2 - v_1) \bullet (v - v_2) \geq 0, \\ z & \text{otherwise,} \end{cases}$$

where

$$z = |v - v_1 - \frac{(v - v_1) \bullet (v_2 - v_1)}{|v_2 - v_1|}|.$$

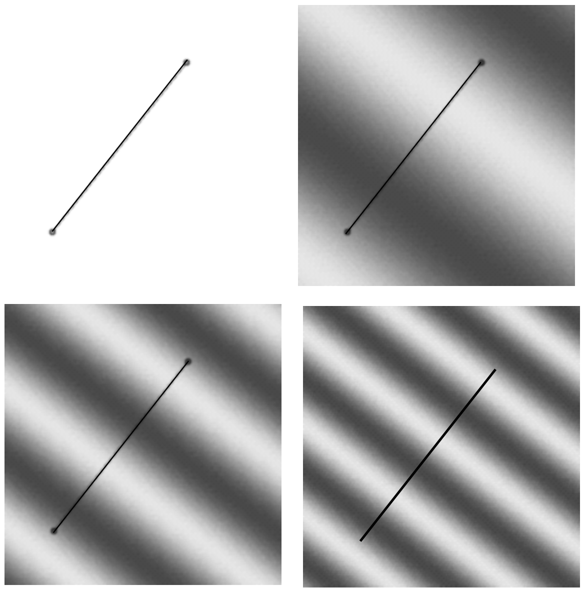

Fig. 1. Periodic function for an oriented line segment. ω values are 2π, 4π and 6π.

Figure 2 shows loci curves which are specified as $d(v) = c$ where c is a positive real number that indicates the distance from the line segment.

Based on the function $d(v)$, we introduce a parameter

$$s = r \, \frac{d(v)}{|v_2 - v_1|}$$

where r is a scale factor. The parameter s is 0 on the line and is 1 at $|v_2 - v_1|/r$. Using this parameter, we introduce the amplitude function

$$a(v) = \begin{cases} (1-s)^3 + 3s(1-s)^2 & \text{if } s \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The amplitude function $a(v)$ is a C^1 function. It takes value 1 on the line segment and 0 for every point whose distance is larger than $|v_2 - v_1|/r$ to the line segment. By multiplying this amplitude function $a(v)$ with periodic function $p(v, t; \omega)$, a flow function that represents orientation, position and length of line segment is

$$f(v, t; \omega) = a(v)p(v, t; \omega).$$

Figure 3 shows the effect of the amplitude.

B. More Than One Line Segment

If there exists more than one line, the problem is to appropriately combine the flow functions $f(v, t; \omega)$ associated with each line. Let n be the number of lines. Then let v_2^i and v_1^i denote the end points, $f_i(v, t; \omega)$ denote the flow function, $h_i(v, t)$ denote the distance function, and r_i denote scaling factor of line i where i is an integer between 1 and n. The amplitude function in the previous section describes a region of influence R_i for each line. This region of influence can be given by the set

$$R_i = \left\{ v \;\mid\; d_i(v) - \frac{|v_2^i - v_1^i|}{r_i} \leq 0 \right\}.$$

Fig. 2. Equidistant curves described by a distance function for an oriented line segment.

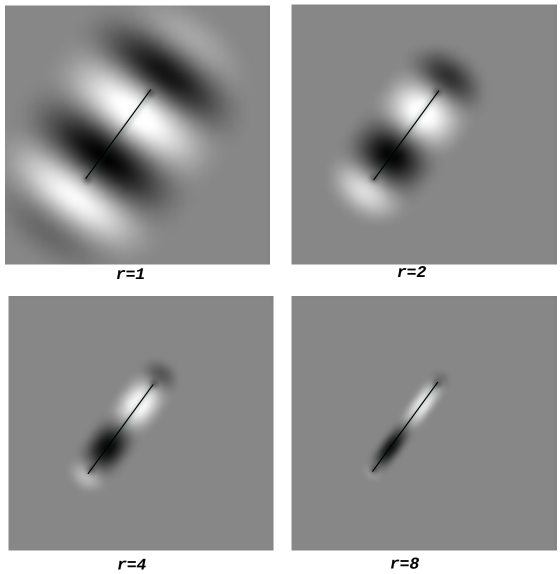

Fig. 3. The effect of amplitude function ($\omega = 4\pi$).

The combined region of influence

$$\mathcal{R} = \bigcup_{i=1}^{n} R_i.$$

of all the lines is given by the union of each region of influence R_i. In implicit surface theory, it is well known that the exact or approximate union operations can be obtained by functional operations such as Rvachev [11] or Ricci operations [12]. Akleman and Chen have recently discovered an effective approximate union operation which is similar to the Ricci operation [1]. Let $g_i(v)$ denote the region of influence function $d_i(v) - \frac{|v_2^i - v_1^i|}{r_i}$. Then, in terms of the new operation, the combined region of influence can be approximated as

$$\mathcal{R} \approx \left\{ v \;\mid\; \frac{-1}{\mu} \log \sum_{i=1}^{n} e^{-\mu \, g(v)} \leq 0 \right\},$$

where μ is a real number that control smoothness of the combined regions and when $\mu \to \infty$, the relation becomes an equality. The gradient of this operation gives another useful operation over gradients of functions, $\nabla d_i(v)$:

$$\nabla \left(\frac{-1}{\mu} \log \sum_{i=1}^{n} e^{-\mu \, g(v)} \right) = \sum_{i=1}^{n} \beta_i(v) \nabla d_i(v)$$

where

$$\beta_i(v) = \frac{e^{-\mu \, g_i(v)}}{\sum_{i=1}^{n} e^{-\mu \, g_i(v)}}.$$

Since

$$\sum_{i=1}^{n} \beta_i(v) = 1,$$

this operation is simply a weighted average of gradient vectors $\nabla d_i(v)$. The weights depend only on v. If $v \in R_j$ but $v \notin R_i \quad \forall i \neq j$ then the result is almost exactly equal to $\nabla d_i(v)$. If v is in the intersection of several regions of influence, the resulting gradient is the weighted average of the gradient vectors associated with these regions of influence. This operation can also be used for combining flow functions. By replacing gradient functions $\nabla d_i(v)$ with flow functions $f_i(v, t)$ we obtain the combined flow function

$$F(v, t; \omega) = \sum_{i=1}^{n} \beta_i(v) f_i(v, t; \omega).$$

The animations can be then computed by appropriately sampling this function. In order to avoid spatial and temporal aliasing, random sampling is needed. We have developed a user interface to create 2D flow animations. By using this interface we have created various 2D flow animations. Figure 4 and 5 show oriented control lines and related frames of cyclic animations. The bottom four images in Figure 4 and 5 are cyclic animation frames for the given set of oriented line segments that are shown on the top row.

Because of the associative and commutative properties of approximate union and gradient interpolation operations, the resulting flow functions $F(v, t)$ can be recomputed in constant time (i.e. it is constant time updateable) after deleting, changing or adding a constant number of line segments [1].

III. Application of Flow Animations to Compressor Visualization

We used the 2D flow animations to improve the quality of compressor visualizations. The compressors are typical of the type found in household refrigerators and air-conditioning systems. There are various types of compressors such as centrifugal, scroll, rolling piston and twin-screw [2]. For heating, refrigerating and air conditioning engineers it is essential to understand and visualize how each type of compressor works. Therefore, the goal of compressor visualization is to illustrate the compression principles for different types of compressors. Unfortunately, it is almost impossible to understand the working principles of compressors just by looking static illustrations. It is

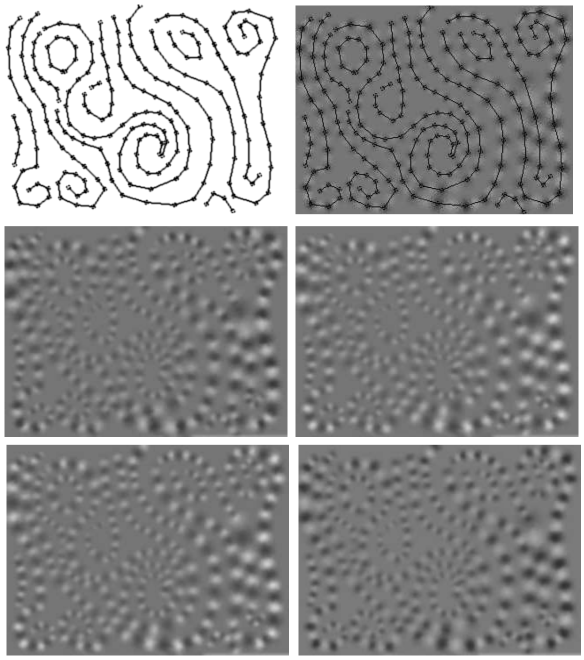

Fig. 4. Oriented lines and related frames of cyclic animation.

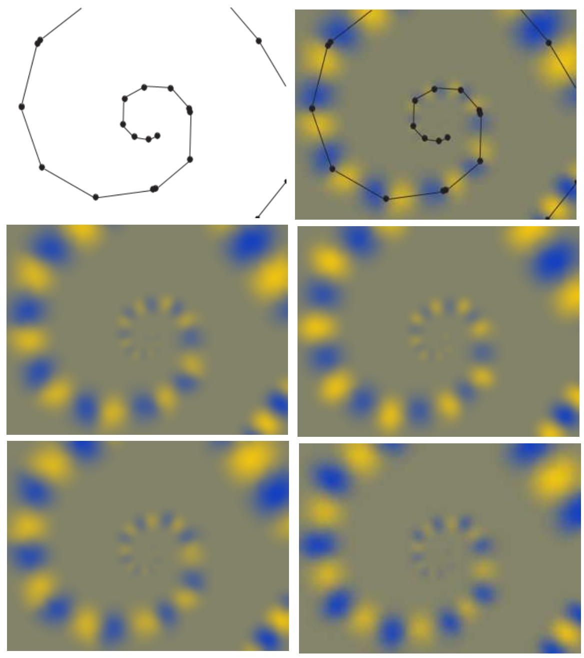

Fig. 5. Another example of cyclic animation.

therefore essential to create animations that show how the individual parts of compressors and the associated fluid move during compression cycles. However, if the animations of the flow were not included, then the viewer is left to imagine how the gases are compressed.

For some compressors, such as scroll, rolling piston and rotary wane, the best staging of the compressor motion is given by parallel projection. For the visualization of such compressors 2D flow animations can be a useful tool. We used 2D flow animations to improve the quality of visualization of rolling piston and rotary wane compressors. Rolling piston compressor uses a roller mounted on the eccentric of a shaft with a single vane or blade suitably positioned in the nonrotating cylindrical housing, generally called the cylinder block. The blade reciprocates in a slot machined in the cylinder block. This reciprocating motion is caused by the eccentrically moving roller. In order to create rolling piston animation, we have developed a simple texture mapped 3D model for the compressor and animated the cylinder blocks by using a commercial modeling and animation package. We have then rendered this animation by using parallel projection. Flow animations are created in our system separately. These two animations are later combined by using a compositing program. Some frames of resulting animation are shown in Figure 6.

As an example of rotary wane compressors we used an eight-bladed compressor. The eight discrete volumes are referred to as cells. In this compressor, a single shaft rotation produces eight distinct compression strokes. Similar to rolling piston case, rotary wane compresser is modeled, animated and rendered in 3D by using a commercial modeling and animation package. Flow animations are created in our system separately. Then the two animations are combined by using a commercial compositing package. Some frames of resulting animation are shown in Figure 7.

IV. Conclusion

In this paper, we have developed a function based approach for user controlled 2D flow animation. Our approach provides a simple algorithm for creating variable-speed flow animations. Since the algorithm is based on constant time updateable operations, recomputation of the flow animation is extremely efficient.

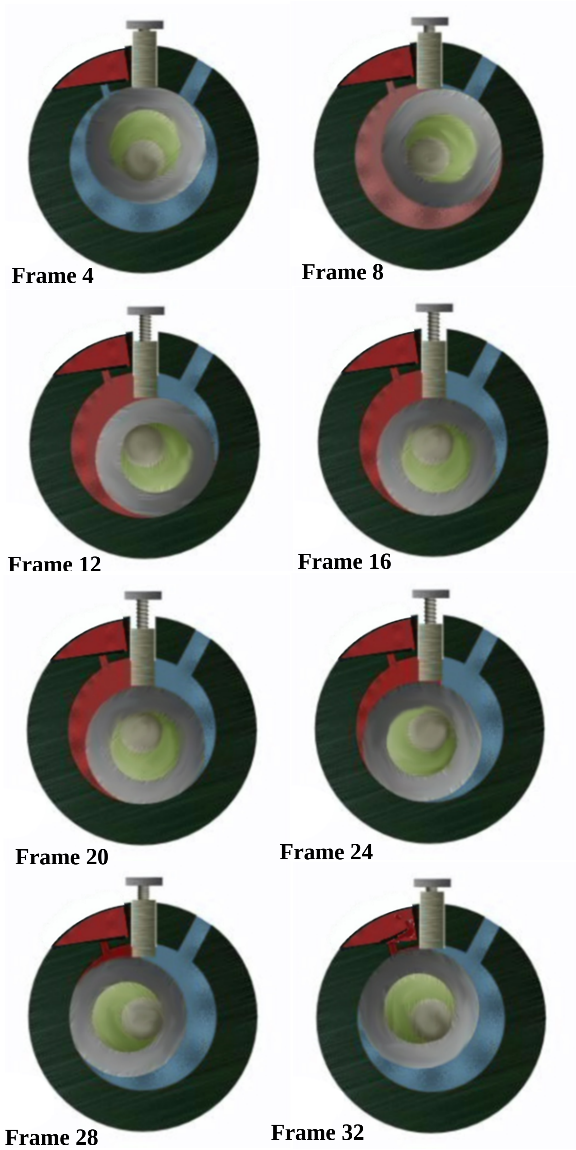

Fig. 6. Sample frames of rolling piston animation.

V. Acknowledgments

Portions of this work were supported by The American Society of Heating Refrigeration and Air Conditioning Engineers (ASHRAE) through research project 1017-RP.

References

[1] E. Akleman and J. Chen, "Constant Time Updateable Operations", *Proceedings of Shape Modeling'99*, 73-80, September 1999.

[2] *ASHRAE, H.V.A.C. Systems and Equipment: Chapter 34 - Compressors,* American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA., 1996.

[3] J. Bloomenthal, "Techniques for implicit modeling", In B. Wyvill and J. Bloomenthal eds., *Modeling and Animating with Implicit Surfaces,* ACM SIGGRAPH'90 Course Notes no. 23, August, 1993.

[4] B. Cabral and L. Leedom, "Imaging Vector Fields using Line Integral Convolution", *Proceedings of ACM SIGGRAPH'93,* pp. 263-272, August, 1993.

[5] L. K. Forsell, "Visualizing Flow over Curvilinear Grid Surfaces Using Line Integral Convolution", *Proceedings of IEEE Visualization'94,* pp. 240-247, October, 1994.

[6] L. Forsell and S. D. Cohen, "Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-Speed Animation and Unsteady Flows", *IEEE Transaction on Visualization and Computer Graphics,* vol. 1, no. 2, pp. 133-141, June, 1995.

[7] W. T. Freeman, E. H. Adelson and D. J. Heeger, "Motion without Movement", *Proceedings of ACM SIGGRAPH'91,* pp. 27-30, July, 1991.

[8] A. Van Gelder ad J. Wilhelms, "Interactive Visualization of Flow Fields", *Proceedings of Workshop on Volume Visualization,* pp. 47-54, 1992.

[9] B. Jobard and W. Lefer, "The Motion Map: Efficient Computation of Steady Flow Animations", *Proceedings of IEEE Visualization'97,* pp. 323-328, October, 1997.

[10] N. Max and B. Becker, "Flow Visualization Using Moving Textures", *Proceedings of ICASE/LaRC Symposium on Visualizing Time-Varying Data,* 1995.

[11] A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko, "Function Representation in Geometric Modeling: Concepts, Im-

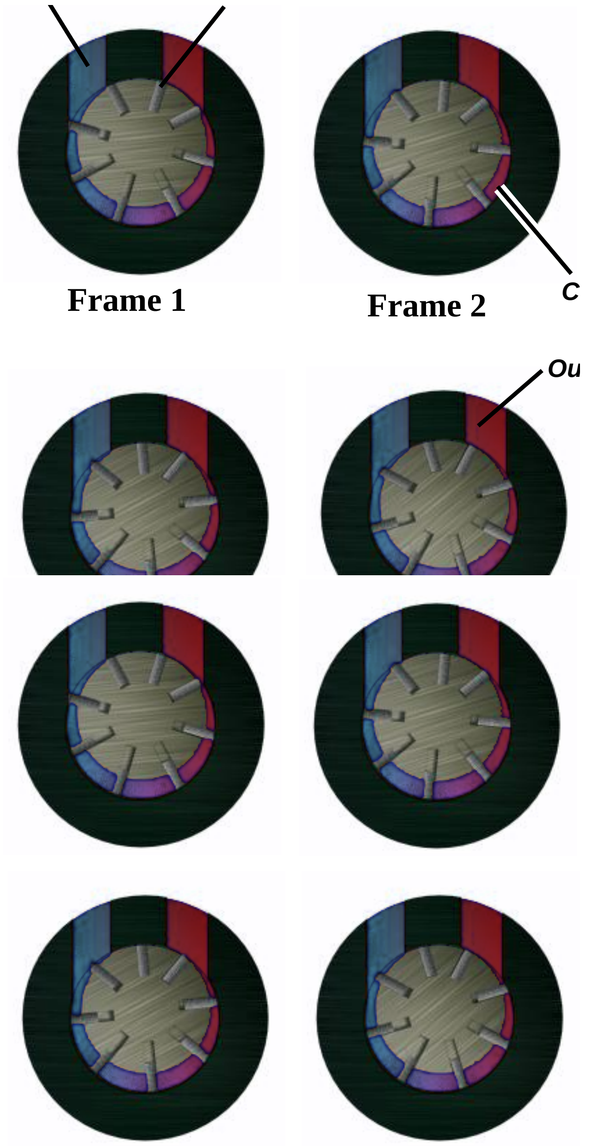

Fig. 7. Sample frames of rotary wane animation.

plementations and Applications", *Visual Computer,* no. 11, pp. 429-446, 1995.

[12] A. Ricci, "A Constructive Geometry for Computer Graphics", *The Computer Journal,* vol 16, no. 2, pp. 157-160, May 1973.

[13] D. Stalling and H. C. Hege, "Fast and Resolution Independent Line Integral Convolution", *Proceedings of ACM SIGGRAPH'95,* pp. 249-256, August, 1995.