Enhanced Line Integral Convolution with Flow Feature Detection

Arthur Okada[1] David Lane[2]

NASA Ames Research Center
M/S T27A-2
Moffett Field, CA 94035

Abstract

The Line Integral Convolution (LIC) method, which blurs white noise textures along a vector field, is an effective way to visualize overall flow patterns in a 2D domain [Cabral & Leedom '93]. The method produces a flow texture image based on the input velocity field defined in the domain. Because of the nature of the algorithm, the texture image tends to be blurry. This sometimes makes it difficult to identify boundaries where flow separation and reattachments occur. We present techniques to enhance LIC texture images and use colored texture images to highlight flow separation and reattachment boundaries. Our techniques have been applied to several flow fields defined in 3D curvilinear multi-block grids and scientists have found the results to be very useful.

1. Introduction

Surface oil flow is a popular technique in experimental flow visualization for observing flow patterns near or on a solid body. Foreign materials such as mixtures of oil and paint are coated over the body surface. When an air stream passes over the surface of the body, the oil paint creates flow patterns on the surface. This technique is particularly effective for observing flow separations and reattachments [Merzkirch '74].

In numerical flow simulations, several techniques have been used to simulate surface oil flows. A method commonly used by computational fluid dynamics (CFD) scientists is to compute a particle trace from each grid point on the surface. Particles are restricted to travel on the grid surface. The simulated surface oil flow generated by this method depends highly on the distribution of the

1. Parametric Technology Corp. 2. MRJ, Inc.

grid points and the length of the particle traces. Although this method can give an overall impression of the flow pattern, the resulting image consists of discontinuous flow lines. Cabral and Leedom introduced the LIC method to create continuous flow textures in a 2D rectilinear grid [Cabral & Leedom '93]. Their technique averages texture map pixel (texel) values along a vector field streamline. The method is local and is effective for visualizing local vector field tangents. The flow textures generated by the LIC method are continuous and are easy to understand. We have applied their technique to visualize some existing flow fields. Common remarks from scientists about the LIC images are that the flow is blurry and that it is difficult to identify boundaries where flow separation and reattachment occur. Another comment is that the images do not have sharp contrast. In this paper, we present techniques to enhance the LIC image texture and propose color texture to highlight flow separation and reattachment boundaries.

2. Related Work

There are some existing visualization techniques that can be used to simulate surface oil flows. Van Wijk introduced spot noise to depict surface oil flows [van Wijk '91]. The output image depends highly on the size, shape, and weight of the spots. For surface oil flows, ellipse shaped spots were found to be effective. Recently, the spot noise technique was enhanced for curvilinear grids using spot bending and spot filtering for low pass frequencies removal [de Leeuw & van Wijk '95]. Forssell and Cohen have extended the LIC technique for curvilinear grids and allowed variable speed animation based on the local velocity magnitude [Forssell & Cohen '95]. Recently, Stalling and Hege proposed a fast LIC method to speed up the computation time [Stalling & Hege '95]. Dovey introduced a technique to draw "glyphs", which are small line segments, at resampled grid points such that the glyphs are uniformly distributed over the grid domain [Dovey '95]. The technique is more effective than traditional arrow plot techniques, which usually draw vectors at each grid point and do not consider the grid density. Although all of the methods discussed above are effective for generating flow textures on the surface, the resulting flow patterns in general do not show flow separation and reattachment boundaries clearly. Recently, de Leeuw et al. have proposed techniques to overlay flow texture generated from the spot noise technique with skin friction lines [de Leeuw '95]. They also overlayed numerical flow textures with experimental photographs from experimental flow results. These techniques are effective, however, sometimes experimental results are not available.

3. Image Enhancements

In this section we present three techniques to enhance flow texture images from the LIC algorithm. First, we propose a double LIC algorithm to improve the delineation of the flow lines. Then, we present two filtering techniques to increase image contrast and remove low frequency of noise. Finally, we describe how to remove texture distortion, which is introduced by the use of LIC on curvilinear grids, by resampling the grid.

3.1 Double LIC

The nature of the LIC algorithm is to average texel values that lie on an integrated path of a streamline in a flow field. When the input texel values of the LIC algorithm are random white noise, the histogram of the LIC image results in a distribution curve with a standard deviation range of approximately 106 to 148 (for 256 gray scale levels). This resulting image is muddy and difficult to clearly see. The ideal output image would consist of a series of long thin distinct contrasting lines, depicting the streamlines. There are several methods which may be used to approach this ideal result.

The resulting image of the LIC algorithm contains dark and light lines that indicate the flow of the vector field. Because these lines are made up of texels created by averaging random noise values, they have a rough undulating quality to their color. A solution to this problem is to alter the length of convolution used in Cabral & Leedom’s LIC algorithm. When longer convolution length’s are used, a greater number of texels on a particular flow line are assigned similar texel color values. But these results come at a performance cost as indicated in [Cabral & Leedom ‘93] where doubling the length increases computation by a factor of four.

Another solution to this problem is to execute the LIC algorithm twice; where the output image of the first execution becomes the input image to the second execution. The second execution of the LIC algorithm creates the flow lines, not by averaging random white noise values, but by averaging like color values that are related to a particular flow line (either dark or light). This results in solid lines that do not undulate as frequently in value. This method results in lines which are more distinct, but at the same time it concentrates the texel values to an even narrower standard deviation range compared to the original LIC output image. This effect can be countered with post-filters applied to the final image.

Performance wise, the second method increases computation time by a factor of two. It is also possible to execute the LIC algorithm three or more times, however, we found no appreciable difference in executing the LIC algorithm more than twice. Since extra computation time is required, we recommend double LIC only.

3.2 Filtering

Two filters have been used to clarify the flow field lines in the texture image: a histogram equalization filter and a 3x3 high-pass filter. The histogram equalization filter moves texel values toward the extremes of the range. It allows for a wider resultant range of multiple levels. Since we have a good idea of the statistical values for the input texture image and we are not interested in a totally equalized histogram for the filtered image, an approximation for the equalization function will do. This function is as follows:

```
if( texel_value > 0 )
   texel_value = texel_value ** power;
else
   texel_value = -( abs( texel_value ) ** power );
```

Here, the function allows for the user to alter the curvature, and ultimately the degree to which this function shifts texel values away from the statistical average. A good range of values for `power` is between 5 and 7.

When analyzing the effects of the histogram equalization filter, we note that the image results have a wider dynamic range of texel values than the original. Yet the image quality still suffers in that distinguishability between individual lines is not necessarily increased. A 3x3 high-pass filter aids in this area where a local region, rather than a pixel by pixel approach to image enhancement is needed. By altering texel values based on neighboring values, we can increase the sharpness of each flow line. We use a 3x3 sized filter for the reason that the LIC algorithm creates flow lines that are quite often 1 texel in width. If a larger filter, say a 5x5 is used, we could quite possibly oversample this local area of the image and decrease the edge enhancing effects of this high-pass filter.

Different combinations of filters and the LIC algorithm are possible. Let *e* denote the histogram equalization filter, *h* denote the 3x3 high pass filter, and *l* denote the LIC algorithm. The sequence *lhe* implies applying the LIC algorithm once, followed by the high pass filter, then the histogram equalization filter. There are varying differences among all the possible combinations. The sequence that we found to produce the best qualitative improvement in image was the sequence *llhe*; which is to perform the LIC algorithm twice, follow by the execution of the 3x3 high-pass filter then the histogram equalization filter.

Figures 1 depicts a sequence of images generated by using the following combinations of techniques: *l, ll, llh,* and *llhe*. The sequence *llhe* gives the best image quality.

3.3 Grid Resampling

The LIC method introduced by [Cabral & Leedom '93] was developed for vector fields sampled on an uniform rectilinear grids. In our flow fields, curvilinear grids were used. After executing the LIC algorithm, the output image is a texture map representing the given flow field in computational space. This image contains evenly sized and spaced texels. But when this image is remapped back onto the curvilinear grid in physical space, a distortion of each texel occurs. This distortion of texels ruins the illusion that these texels, combined together, create flow lines.

The root of this problem lies in the location of the grid lines. When studying flow fields, grid lines are often closely concentrated at areas of interest; areas where large amounts of information occur such as at a wing's leading edge. Where information content is low, grid lines are usually widely spaced. This leads to efficient memory usage for the data, but for visualization using the LIC algorithm, this leads to distortion of texels in size and aspect ratio. A restructuring of the grid lines would ease this situation by redistributing the texels evenly throughout the grid in physical space.

In our implementation, a user supplied spacing variable is used to determine the approximate spacing between the new grid lines. The lengths of the two end grid lines and the center grid line (weighted twice) is averaged to get an approximate size for the grid's dimension in one direction. This averaged length is then divided by the spacing variable. The ceiling of this result then becomes the number of cells we will have in this direction. Using the longest of the three grid lines that we previously used, we divide this line by the resultant number of cells to get the actual spacing.

The resampled grid points can be interpolated from the original grid. Similarly, the velocity field is interpolated at the new grid points. Most often the grid is resampled at a higher resolution than the original grid such that there is very little loss of information from the original velocity field. Because the resampled grid lines are evenly spaced, the LIC image resulting from this new grid, when mapped back onto the grid, will have texels of similar size ranges and aspect ratios closer to 1:1.

For efficiency and accuracy, it is desirable to remap the resultant texture map onto the original grid because of two factors. First, the resampled grid often contains hundreds of times more grid points compared to the original grid. For efficiency, it is better to store only the original grid points rather than storing the new grid nodes. Secondly, in the resampled grid there may be areas where grid shape accuracy is lost, such as at leading edges where a line may disappear, affecting the overall shape of the grid. We can easily remap the texture map back to the original grid by using a set of parametric values of original grid lines to the newly spaced grid lines. This allows many texels to be contained within a single large grid cell, while one or a fraction of a texel may be assigned to multiple grid cells of minute dimensions.

4. Flow Feature Detection

Surface oil flows are often used by experimental flow scientists to observe flow separations and reattachments. Unfortunately, most existing visualization techniques do not highlight these flow features clearly. We propose two techniques to highlight these flow features. Firstly, coloring the flow texture based on velocity direction allows changes in flow direction to be easily seen. Secondly, by coloring the flow texture based on velocity angle, it is easy to identify boundaries of flow separations and reattachments.

4.1 Color by velocity directions

To detect changes in flow direction, the texture image can be colored by the velocity direction in computational space. This would create regions of the flow grouped by their directions. For example, a different color band is assigned to a range of velocity directions. All velocity vectors within a particular range will receive the same color. An advantage of assigning color by velocity direction is that vectors with similar directions are grouped into regions that are easy to see. Flow separation occurs when the flow converges to a common region. Flow reattachment occurs when the flow diverges from a common region. In the monochrome LIC images, these features are not very prominent and are sometimes difficult to see. However, by grouping the vectors by direction, these features are easier to see.

To add color to the our program, several color schemes were tested. For any given vector, a corresponding scalar is obtained representing direction. This scalar is then used as an index to a color lookup table. In conjunction with the grey scale value given by the LIC algorithm, we create a new RGB value for each texel. With the following coloring scheme (RGB->B), we vary the RGB value from the indexed color to black:

```
RGB = ( LIC / 255 ) * color,

where LIC is the texel value resulting from the LIC algorithm and color
is a (r,g,b) vector;
```

Another color scheme (W->RGB->B) is to vary the RGB value from white, to the indexed color, to black:

```
if( LIC < 128 )
   RGB = ( LIC / 127 ) * color;
else
   RGB = color + ( 255 - color ) * ( ( LIC - 127 ) / 128 );
```

4.2 Color by velocity angles

Although assigning color to the flow textures by velocity direction helps to visualize different flow regions, flow separation lines and reattachment lines sometimes are sometimes still difficult to perceive. This is in part due to the fact that not every color boundary is a separation or reattachment line. In this method, we propose to color the image at regions near flow separation and reattachment while only using gray colors at the rest of the image. By detecting regions where the neighboring velocity angle is high, we can highlight these flow features. To calculate the color for each texel, we find the degree of separation or attachment by the dot product between vectors of neighboring adjacent areas and itself. The minimum value, of all these dot products, is used as an index to the color look up table.

In detecting separation and attachment lines, using simply the dot product of each texel with its neighboring texel may overlook separation areas or falsely add separation areas. This anomaly may be caused by several factors. Separation and attachment lines often have varying rates of change in direction. Some lines slowly converge together, and as such, a 3x3 algorithm to detect these lines might completely miss such a slowly converging line. If an algorithm with a larger filter is used, a flow with narrow rapidly changing flow separation lines may also be ignored. With a larger filter, possible false reading might also occur as several separation and attachment lines may be contained within the filter size. If several lines are found within the filter size, a false minimum value of the resultant dot products is likely to occur.

In our algorithm, we choose to group vectors into 3x3 texel areas, then create an average for each group. With these averaged vectors, we calculate the minimum dot products between itself and its 8 neighboring groups. We use the resultant color for each of the nine texels within the group, but using each texel's own LIC output value, to determine the final color for the texel. By applying this low-pass filtering technique, detection of slower forming separations and attachments lines are enhanced, while avoiding the possible false minimum dot product readings. The success of this technique largely depends on the data and the size of the vector groupings. Again, if the flow is changing over a large region, a larger filter grouping may be required.

We have also tried to use distinct colors to differentiate highlighted flow separation versus highlighted flow reattachment lines. This can be achieve by computing additional information at each texel position. One possible approach is to determine if the flow is leaving the surface or approaching the surface. If the flow is leaving the surface, then it implies flow separation. Conversely, if the flow is approaching the surface, then it implies flow reattachment. Therefore, by examining the direction of the velocity vector normal to the grid surface, we can distinguish between these two flow features.

5. Animated Flow LIC (AFLIC)

We have developed an object oriented visualization system called the Animated Flow LIC (AFLIC), which supports curvilinear multi-block grids. Other features of AFLIC allow for moving grids, multiple grid surface definitions and supplemental animation sequence creation. During program execution, the 3D grid and velocity files are first read in. The grid and velocity are then reduced and constrained to the desired grid surface's two dimensional limits and plane index. The grid resampling algorithm is then used to construct a new grid and velocity data set. At this point, the velocity field is converted from physical space to computational space. These computational space vectors are then used by the LIC algorithm to create the texture map. Any additional iterations of the LIC algorithm or any filters are then applied to the texture map. These steps are repeated for as many solution time steps as desired. The texture maps are then saved into files that can be read by a viewing program at a later time.

In a sample data set, 83% of processing time is spent convolving along the line integral. Decreasing the computation time in the LIC algorithm would greatly benefit the AFLIC program's performance. Recently Stalling and Hege improve the LIC algorithm based on the fact that several texels might share streamline information that one line integral convolution would create. This sharing of information reduces the overall number of computations required [Stalling & Hege'95].

6. Case Studies

We have applied the image enhancement and flow feature detection techniques to several numerical flow fields. In this section, we present the results from two numerical flow simulations: a delta wing and an ogive cylinder body. Figure 2 depicts the geometry models used for the two simulations. For the first simulation, the scientists want to analyze surface flow separation lines and periodic motion of vortex breakdowns near the trailing edge of the delta wing [Chaderjian & Schiff '95]. All of the figures depicted in this section use the following sequence of algorithms discussed in Section 3: resampling grid to a higher resolution, double LIC, high pass filter, and histogram equalization. Figure 3 depicts simulated surface oil flow where the texture is colored by flow direction and the RGB->B coloring scheme is used. Figure 4 depicts the texture, colored also by flow direction, but with the W->RGB->B coloring scheme. Comparing Figures 3 and 4, it is clear that Figure 4 shows more contrast. The change in color gives a good indication of change of flow direction. However, not all flow separations and reattachments are obvious because all color boundaries do not indicate these flow features. Figure 5 depicts the same flow field except the textures are colored by velocity angles with the RGB->B coloring scheme. Using only a red color scale to shade boundaries near flow separation and reattachment and gray values for the rest of the image, visibility of these boundaries are increased. Figure 6 depicts delta wing also colored by velocity angle. In addition, velocity directions normal to the surface is used to distinguish between flow separation and flow reattachment. In the leading edge of the delta wing, the green denotes flow separation lines and light brown-red denotes flow reattachment lines.

The second flow analysis is for a simulated flow about an ogive cylinder body. Figure 7 depicts numerical surface flow with texture color based on velocity direction. The magenta band along the cylinder is due to the shedding of boundary layers. An animation of this visualization reveals the boundary layer shedding even more dramatically. Figure 8 depicts the same flow but the texture is colored by velocity angle. Using the RGB->B coloring scheme, the flow separation and reattachment boundaries are also easy to detect.

7. Summary and Conclusions

Several techniques are presented to enhance LIC images and to highlight flow separation and reattachment in numerical flow fields. The capability to identify and distinguish these flow features can assist scientists in the analysis of their flow simulations. We found colored textures based on neighboring velocity angles to be the most effective in this type of flow feature detection. Further research can be performed to investigate flow feature detection in unsteady flows. Another research area of interest is to extend the LIC method for time-dependent (unsteady) flows. A potential problem with unsteady flows is that streaklines instead of streamlines should be used. However, streaklines often cross each other, resulting with an indiscernible spaghetti like pattern. Using the LIC algorithm, neighboring texel values will have no correlation to each other resulting in a noise filled pattern.

Acknowledgments

We thank Neal Chaderjian and Scott Murman for provided us with the flow data sets and for evaluated our preliminary results with LIC. We also thank Michael Gerald-Yamasaki, David Kenwright, and David Yip for their helpful suggestions and all our other colleagues in the NAS Division for providing ideas and information. This work was supported by NASA under contract NAS 2-14303.

References

[1] W. Merzkirch, Flow Visualization, Academic Press, New York, 1974.

[2] B. Cabral and L. Leedom, Imaging vector fields using line integral convolution, Proceedings SIGGRAPH ‘93, Anaheim, California, August 1993, pp. 263-272.

[3] van Wijk, Spot noise – texture synthesis for data visualization, Proceedings SIGGRAPH ‘91, Las Vegas, Nevada, July 1991, pp. 309-318.

[4] de Leeuw and van Wijk, Enhanced spot noise for vector field visualization, Proceedings Visualization ‘95, Atlanta, Georgia, October 1995, pp. 233-239.

[5] L. Forssell and S. Cohen, Using line integral convolution for flow visualization: curvilinear grids, variable-speed animation, and unsteady flows, IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 2, June 1995, pp. 133-141.

[6] D. Stalling and H. Hege, Fast and resolution independent line integral convolution, Proceedings SIGGRAPH ‘95, Los Angeles, California, August 1995, pp. 249-256.

[7] D. Dovey, Vector plots for irregular grids, Proceedings Visualization ‘95, Atlanta, Georgia, October 1995, pp. 248-253.

[8] W. de Leeuw, H.-G. Pagendarm, F. Post, B. Walter, Visual simulation of experimental oil flow visualization by spot noise images from numerical simulation, Sixth Eurographics Workshop on Visualization in Scientific Computing, Chia, Italy, May 1995, to appear.

[9] N. Chaderjian and L. Schiff, Navier-Stokes analysis of a delta wing in static and dynamic roll, AIAA Paper 95-1868.

Figure Captions

Fig. 1 (A) single LIC, (B) double LIC, (C) double LIC with high-pass filter, (d) double LIC with high-pass & equalization filter.

Fig. 2 Geometric models of a delta wing and an ogive cylinder.

Fig. 3 Delta wing colored by velocity direction using color scheme: from indexed color to black.

Fig. 4 Delta wing colored by velocity direction using color scheme: from white to indexed color to black.

Fig. 5 Delta wing colored by velocity angle.

Fig. 6 Delta wing colored by velocity angle and velocity directions normal to surface.

Fig. 7 Ogive cylinder colored by velocity direction.

Fig. 8 Ogive cylinder colored by velocity angle