GLIC: AN INTERACTIVE SOFTWARE TOOL FOR VISUALIZING SURFACE FLOWS

Han-Wei Shen* David L. Kao† Ling-Jen Chiang‡ Aleksandra Kuswik§

Abstract

This paper presents an interactive visualization software, called GLIC (Graphical Line Integral Convolution), for visualizing data from steady and unsteady CFD simulations. The system features a new type of visualization technique, called Line Integral Convolution (LIC), which converts flow data into images and animations to capture complicated flow behavior and to help scientists detect important flow features. GLIC contains a flexible graphical user interface and an efficient multi-threaded software environment in which users can interactively specify regions of interest and effectively steer visualization with minimum effort. GLIC is ideal for visualizing data from large-scale CFD simulations with complex model geometries.

Introduction

Effective visualization of computer-generated flow field data is challenging due to the difficulties in finding suitable graphical icons for depicting flow directions and highlighting important features. To analyze data from large-scale numerical simulations with complicated model geometries, the techniques used by existing visualization software, such as vector plots or particle tracing, often fail to provide sufficient information for characterizing the underlying flow phenomena. The problems mainly come from the difficulties in selecting the seed locations from which the vectors or particles are introduced. To create effective visualization, these techniques often require the user to have a priori knowledge about the data, or the selections of the seed locations are done on a trial-and-error basis. In many cases, important flow features can be missed rather easily.

This paper presents an interactive visualization software, called GLIC (Graphical Line Integral Convolution), for visualizing vector field data from steady and unsteady CFD simulations. The system features a new type of visualization technique, called Line Integral Convolution (LIC) [1, 2] which can generate texture patterns similar to the surface oil flows commonly produced in wind tunnel experiments. Unlike the conventional techniques, GLIC does not require users to specify seed locations for tracing particles or attaching vector plots. Instead, the flow textures are automatically computed and mapped onto the model surfaces that are desired to be visualized. To facilitate flexible specifications of regions of interest from the underlying model, GLIC provides a graphical user interface (GUI) for interactive interrogation and definition of model surfaces and texture resolutions. In addition, GLIC executes in an efficient multi-threaded software environment, which allows simultaneous executions of user event handling and LIC computation. GLIC is a new software tool that adopts state-of-the-art flow visualization techniques to analyze surface flows in large-scale CFD simulations with complex model geometries.

In the following, we first give a brief overview of the LIC algorithms that are used by GLIC and then describe the functionality and the user interface of the GLIC software tool. We present examples of CFD visualization using GLIC.

Background

This section presents a brief overview of the LIC algorithm and its extension, called UFLIC (Unsteady Flow LIC), for visualizing unsteady flows. For more information, survey articles can be found in [3, 4], and detailed descriptions of the LIC and UFLIC algorithms can be found in [1, 2].

The Line Integral Convolution method is a texture synthesis technique that can be used to visualize two-dimensional vector field data. Taking as the input a

---

*MRJ/NASA Ames Research Center, Mail Stop T27A-2, Moffett Field, CA 94035 (hwshen@nas.nasa.gov)

†NASA Ames Research Center, Mail Stop T27A-2, Moffett Field, CA 94035 (davidkao@nas.nasa.gov)

‡MRJ/NASA Ames Research Center, Mail Stop T27A-1, Moffett Field, CA 94035 (lchiang@nas.nasa.gov)

§Department of Computer Science and Engineering, University of California Santa Cruz, Santa Cruz, CA (kuswik@cse.ucsc.edu)

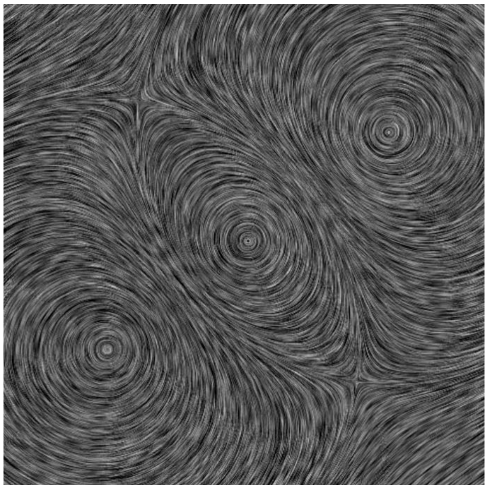


Figure 1: Visualization of a 2D vector field using LIC

vector field and a white noise image with the same resolution as the vector field, LIC computes convolution of the input noise image using the following algorithm: For each pixel, streamlines in both the positive and negative directions are first calculated. The pixel's convolution result is computed by weighted-averaging the image values of the pixels along the streamline paths. The convolution of the entire image results in the fact that the intensity values of the pixels along each streamline are strongly correlated so the directions of the flow field can be easily visualized. Fig. 1 shows an example of an LIC image generated from a two-dimensional vector field.

Although LIC is an image synthesis method, which requires the underlying vector field to have a uniform Cartesian grid, it can be modified and applied to curvilinear gridded data. This is done by computing LIC for the input field in *computational space*, where a curvilinear surface mesh corresponds to a regular Cartesian grid. After the LIC computation is complete, the resulting image is then mapped back to the surface in physical space using texture mapping techniques.

The original LIC algorithm is primarily used for visualizing steady flows. For unsteady flow data, Shen and Kao [2] proposed a new convolution algorithm, called *UFLIC*, that simulates the advection of flow traces globally in unsteady flow fields. Starting from a white noise image as the input texture, the UFLIC algorithm successively advects the texture to create a sequence of flow images. To achieve this, a new convolution method, called the *Time-Accurate Value Scattering* scheme, which incorporates time into the convolution, is used. This time-accurate value scattering convolution scheme, driven by a *Successive Feed-Forward* process, iteratively takes the convolution output from the previous step as the texture input for the next convolution to produce new flow textures. The UFLIC algorithm can effectively capture complicated time-dependent flow phenomena and produce animations with spatial and temporal coherence for tracking dynamic flow features over time. Fig. 2 shows an image snapshot from an animation sequence generated by UFLIC.

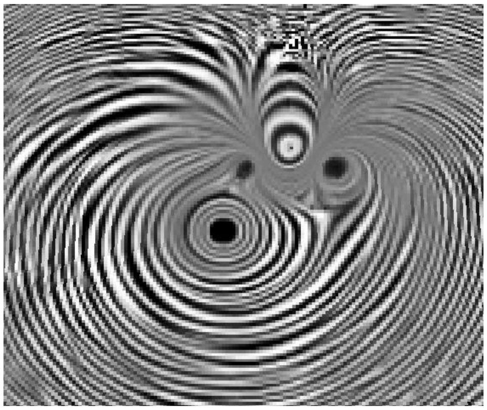


Figure 2: A 2D unsteady flow visualization using UFLIC

We have briefly reviewed the visualization algorithms that are used by GLIC to visualize surface flows. In the following, we describe the functionality and the user interface of the GLIC software.

The Software

The main purpose of the GLIC software is to provide the user with a simple user interface to the LIC algorithms mentioned above for visualizing surface flows. To realize this goal, a graphical user interface, written in the scripting language Tcl/Tk [5] (developed by John K. Ousterhout, University of California, Berkeley), is built into the GLIC program. The interface allows the user to explore the model geometry, launch the LIC computation, and review the visualization results interactively. In general, a typical session of visualization using GLIC consists of the following main steps:

- Describe and read the data files

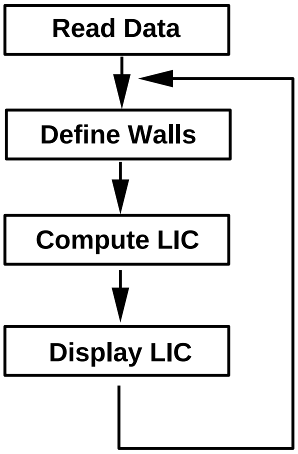
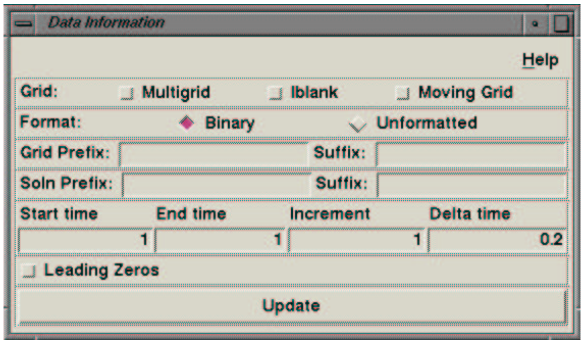


Figure 3: GLIC process pipeline



Grid: ☐ Multigrid ☐ Iblank ☐ Moving Grid
Format: ● Binary ○ Unformatted
Grid Prefix: Suffix:
Soln Prefix: Suffix:
Start time 1
End time 1
Increment 1
Delta time 0.2
☐ Leading Zeros

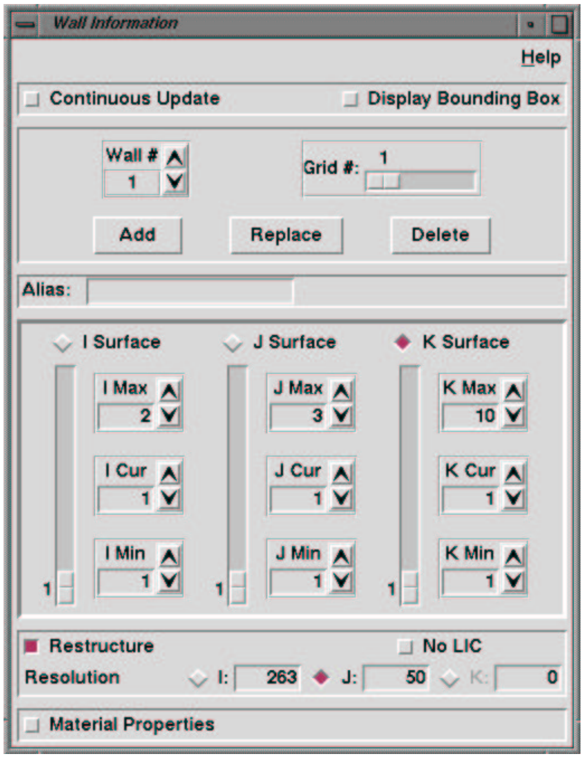
Figure 4: GLIC Data Panel

- Define walls
- Compute LIC
- Visualize walls with LIC textures

Fig. 3 illustrates the pipeline.

GLIC accepts PLOT3D [6] curvilinear gridded data files. To describe the input flow field, a *Data Panel* user interface is provided. Using this interface, the user can specify that the input field is defined on a single- or multi-zone mesh, has steady or time-dependent solutions, and is represented in the binary or Fortran unformatted file format. For the user's convenience, an on-line help menu is available. Fig. 4 shows the Data Panel graphical user interface.

GLIC computes LIC textures on grid surfaces. In GLIC, a grid surface is called a *wall*. GLIC provides a *Wall Panel* user interface to facilitate interactive selections of walls in which LIC textures are computed.



☐ Continuous Update ☐ Display Bounding Box
Wall # 1
Grid #: 1
Alias:
○ I Surface ○ J Surface ● K Surface
I Max 2
I Cur 1
I Min 1
J Max 3
J Cur 1
J Min 1
K Max 10
K Cur 1
K Min 1
☑ Restructure ☐ No LIC
Resolution ○ I: 263 ● J: 50 ○ K: 0
☐ Material Properties

Figure 5: GLIC Wall Panel

Using the wall panel, users can specify walls within any zone in the mesh and can specify the intended texture resolutions that will be used to compute LIC. The degree of control over the texture resolution is useful when the user wants to preview the results at a low resolution before starting a more accurate, but much slower, computation. The Wall Panel is directly linked to a graphical *Viewer* window implemented using the OpenGL graphics library [7]. The walls defined in the Wall Panel are displayed in this Viewer window. From the Wall Panel, the user can interactively modify the walls and visualize the resulting geometries in the Viewer window. The user can also modify the wall's color and material attributes on the fly. For a simulation model with a complex grid geometry, the user can interactively visualize the walls that are selected before the LIC computation starts. Fig. 5 is an example of the Wall Panel.

The LIC computation is controlled by the *LIC Panel*, where the user can select the instantaneous or the time-dependent LIC method as the underlying algorithm. The instantaneous method is an implementation of the original method proposed by Cabral and Leedom [1], and the time-dependent method is

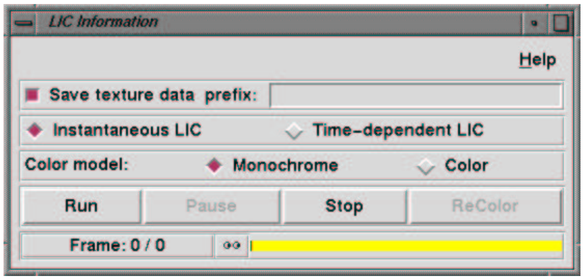


Figure 6: GLIC LIC Panel

the UFLIC algorithm proposed by Shen and Kao [2]. When the input data is a steady flow field, the instantaneous algorithm should be selected. If the input field is a time-dependent flow field, the user can select either the instantaneous or the time-dependent algorithms. Using the instantaneous LIC method is analogous to using streamlines to visualize unsteady flows, while the time-dependent LIC corresponds to the streakline technique. Both types of the LIC textures can be be color-mapped, provided that a scalar function from the solution is specified by the user. Currently, GLIC can compute all the scalar functions that are supported by PLOT3D [6]. In addition, the user can also invoke a color map editor to modify the color map. Fig. 6 shows the LIC panel.

GLIC is not only a program for computing the LIC textures, but is also used as a graphical viewer to animate the LIC textures being computed. The graphical viewer built into the GLIC can control the speed of the animation, save and restore the viewing parameters for positioning the simulation models, adjust the locations of the lights, and dump the display directly into image files. In addition, the graphical viewer can display other important flow features, such as vortex cores or particle traces, which are extracted by UFAT (the Unsteady Flow Analysis Toolkit) [8], to create a more comprehensive visualization. Fig. 7 shows the graphical viewer window.

An important feature of GLIC is that it is a multithreaded program, i.e., the whole program consists of several processes running independently. In GLIC, the tasks of handling user events and computing LIC are executed in different threads. In this way, the user can continuously interact with the program. For example, while the LIC algorithm is computing, the user can still select menu items or rotate the model geometry, check the visualization results, pause the LIC computation, reset the program parameters, or even terminate the LIC process at any time. Another useful feature of GLIC is that the user can save the LIC textures out to files. When analyzing a long sequence of unsteady flow data, the user may decide that the graphical user interface is no longer needed. In this case, he or she can let the program run and save the results for later investigation.

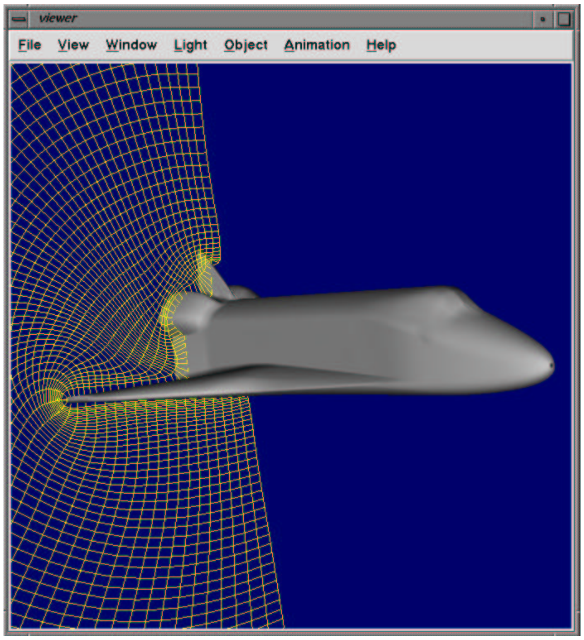


Figure 7: GLIC Viewer Window

GLIC provides the user the capability of scripting; that is, all the parameters specified using the graphical user interface can be saved into a *meta file*. The meta file is retrieved the next time when the user needs to visualize the same data set so that the process of initializing data and walls can be bypassed.

Results and Summary

We have used GLIC to analyze surface flows from several CFD flow datasets. Shown in Fig. 8 is the typical display of an interactive session using GLIC. The following user panels are shown in the figure: Data, Wall, LIC, and Viewer. Using the flow data from a simulation of the F/A-18 fighter jet, surface flows are computed and shown in the Viewer window. To supplement the surface flow visualization generated by the LIC algorithm, three-dimensional particles traces are imported into the GLIC viewer window. The graphics help the scientist to capture complicated flow behavior and detect important flow features. In addition, GLIC'S flexible graphical user interface and efficient multi-threaded software environment allow the user to interactively specify regions of interest and effectively steer the visualization with a minimum amount of effort. GLIC is currently implemented on SGI graphical workstations. The beta version of GLIC has been released.

Acknowledgments

This work was supported in part by NASA contract NAS2-14303. We would like to thank Ken Gee, Ravi Samtaney for providing their data sets.

References

[1] B. Cabral and C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of SIGGRAPH '93*, pages 263–270. ACM SIGGRAPH, 1993.

[2] H.-W. Shen and D.L Kao. A new line integral convolution algorithm for visualizing time-varying flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):98–108, June 1998.

[3] H.-W. Shen. Using line integral convolution to visualize vector fields. In *Computers in Physics*, pages 474–478. American Institute of Physics, October 1997.

[4] D.L. Kao and H.-W. Shen. Numerical surface flow visualization. In *36th AIAA Aerospace Sciences Meeting and Exhibits, AIAA-98-0076*, January 1998.

[5] J.K. Ousterhout. *Tcl and the Tk Toolkit.* Addison-Wesley Publishing Company, 1994.

[6] P. Walatka, P. Buning, L. Pierce, and P. Elson. PLOT3D user's manual. *NASA Technical Memoradum 10167*, March 1990.

[7] J. Neider, T. Davis and M. Woo. *OpenGL Programming Guide.* Addison-Wesley Publishing Company, 1993.

[8] D.A. Lane. UFAT - A particle tracer for time-dependent flow fields. In *Proceedings of Visualization '94*, pages 257–264. IEEE Computer Society Press, Los Alamitos, CA, 1994.