A 3D FLUID-FLOW VISUALIZER FOR ENTRY LEVEL COMPUTERS

A. Sanna[†] B. Montrucchio[†] R. Arina[‡] L. Massasso[‡]

[†]Dipartimento di Automatica e Informatica,
[‡]Dipartimento di Ingegneria Aeronautica e Spaziale,
Politecnico di Torino
corso Duca degli Abruzzi 24,
10129 Torino (Italy)
email: sanna@polito.it

ABSTRACT

In computational fluid dynamics (CFD), visualization is a frequently used tool for data evaluation and qualitative comparison to flow visualizations originating from experiments; moreover, the visualization of three-dimensional fluid flows can be a very important teaching instrument. In particular, it has been proved that streaklines and timelines are effective visualization techniques for depicting time-dependent phenomena such as vortical flows (vortex shedding, vortex breakdown, an so on), moving shock waves and turbulent flows. Often, the visualization of unsteady flows requires a high computational power available only by using high-end workstations. In this paper we present a platform-independent software for particle tracing by pathlines, streaklines, and timelines which can be used in an interactive way on entry level computers; moreover, we propose an innovative texture-based visualization approach we called TOSL (Thick Oriented Streamline).

Keywords: Scientific visualization, flow visualization, vector field visualization.

1 INTRODUCTION

Visualizing 3D fluid-flows remains one of the most challenging applications in scientific visualization. The visualization is often used in computational fluid dynamics (CFD) for understanding the flow characteristics and comparing the numerical simulations with experimental data. Of particular interest is the visualization of time-dependent phenomena such as three-dimensional unsteady flows and several techniques have been proved to be effective for depicting vortical flows, such as vortex shedding and vortex breakdown, moving shock waves, or to elucidate the complex topological nature of turbulent flows. Basically, these methodologies are based on particle tracing [Lane96a] and [Niels97a]; in particular, pathlines, streaklines, and timelines can reveal vortex dynamics more clearly than instantaneous flow visualization approaches (contour lines, vector plots, and volume slices).

The visualization of complex 3D fluid

flows produces images with heavily overlapping geometry. For this reason, powerful workstations are often employed in order to allow the user stereo cues and interactive and intuitive changes of the point of view; in this way, the user can get a better impression of the structure of the 3D flow. In order to do this, fast texture-based visualization approaches are adopted; these methodologies can take advantage from the hardware acceleration of graphics workstations providing, often, real-time performances. In particular, Line Integral Convolution (LIC), introduced by Cabral and Leedom in [Cabra93a] has been proved to be an effective technique for representing directional information via patterns of correlation in a texture. This approach has been further investigated and developed in [Wijk93a], [Forse95a], [Wegen97a], and [Shen98a]; a survey of recent advances in visualizing 3D fluid flows with LIC can be found in [Inter98a]. While these techniques produce impressive visual results, they are not applicable for a real-time use on entry level computers such as PCs without hardware acceleration.

New approaches for real-time 3D flow visualization have been proposed in [Fuhrm98a]; in particular, a new texture mapping technique is presented. Moreover, two new instruments called magic lenses and magic boxes allow the user a better inspection of the 3D structure. Anyway, real-time performances are obtained only when a hardware acceleration for the texture mapping is available.

Some systems have already been developed for steady and unsteady flow visualization; Plot3D [Bunin85a], Plot4D [Smith89a], CM5 [Jespe91a] and pV3 [Haime94a] are examples of software for visualizing 3D structures and animations. Unfortunately, the performance of all these software is strongly related to the employment of accelerated hardware, therefore, they are not well suited to be used on entry level computers usually employed in teaching activities.

This paper presents a platform- independent software for particle tracing by pathlines, streaklines, and timelines which can be used in an interactive way on entry level computers. In order to guarantee the software portability, the interface has been entirely developed with the OpenGL [Neide93a]. This software can be used as a teaching instrument in laboratories equipped with Pentium PCs not provided of accelerated graphical hardware. Moreover, this software supports an effective and cheap system for the stereoscopic visualization in order to provide the user a better comprehension of a 3D flow; the system of stereoscopic visualization is based on interlaced liquid crystal shutter glasses. The software is able to visualize steady as well as unsteady flow data on uniform cartesian grids which can be visualized together the particle traces.

The content of the paper is as follow: a brief background on the particle tracing in presented in section 2, while the objectives of the paper are outlined in section 3. We describe in the detail our software in section 4, and in section 5 we present some examples.

2 BACKGROUND

An effective way to visualize unsteady flows it to compute particle traces. Three types of traces can be computed: pathlines, streaklines, and timelines. A pathline shows the trajectory of a single particle released from one fixed location, called seed location or insertion point. A streakline is a line joining the positions, at an instant in time, of all particles which have been previously released from the same seed location. In order to simulate a streakline the particles have to be continuously released from an insertion point at each time step; for streakline computations, particles are released from user specified seed locations and then they are traced through the time steps. A timeline is a line joining all the particles which have been released at the same time from different insertion points; timelines can be simulated by releasing simultaneously a line of particles at some time interval.

For steady flows, pathlines and streaklines are identical, while for unsteady flows streaklines can reveal several information not visible by tracing pathlines; for instance, vortices can be visualized by streaklines and timelines but not by pathlines. Furthermore, animated streaklines and timelines effectively show the development of the flow phenomena in time.

3 OBJECTIVES

The main objective of the proposed work is to develop a tool for visualizing 3D steady and unsteady flows; this software has to guarantee the user a good interactivity even if used on an entry level computer equipped by a Pentium Processor. Moreover, a set of guide lines inspired the design and the implementation of our software:

1. Portability. The proposed software has to be utilizable on the most common platforms: Windows 95/NT, Linux, and all the other Unix variants both for PC and workstation. Moreover, it has to take advantage of the hardware acceleration if available.
2. Effectiveness. The proposed software has to allow the user to visualize the fluid flows by tracing: pathlines, streaklines, and timelines. Moreover, the user has to be able to control both the time step and the number and the location of the seed points. Furthermore, the software has to provide the user also the classical instruments of texture-based investigation.
3. Usability. The user has to be able to intuitively and interactively change the point of view in order to get a better comprehension of the 3D structure. Furthermore, special tools such as volume slices (cutting planes) and 2D projections of the structure have to be available.
4. Feeling of immersion. The software has to support a stereoscopic visualization device.
5. Compatibility. The program has to be able to save the simulations in a standard format in order to allow the user future 3D data investigations.
6. Flexibility. The software has to allow the user to set the complexity of the 3D geometry to be visualized to adapt the work load to the available computational power.

In section 4 we describe in the details how all these issues have been tackled. In the present work, the flow field data are distributed on uniform rectilinear grids. Being our primary interest to show the ability of the visualization software in capturing the main flow field characteristics, in the examples we avoid further complications due to the geometry. However the extension to general curvilinear coordinates, and unstructured grids, is straightforward, and it does not require major modifications to the software.

4 THE SOFTWARE

One of major goals of our work is to design and implement a platform-independent software; in order to do this, the code is entirely written in C and both the interface with the user and the rendering part is based on the graphics library OpenGL. The OpenGL allows to use the program on PCs running Windows 95/NT, or Unix variants such as Linux, as well as on workstations such as Silicon Graphics, Digital and so on; moreover, high-end PCs and graphical workstations often have ac-

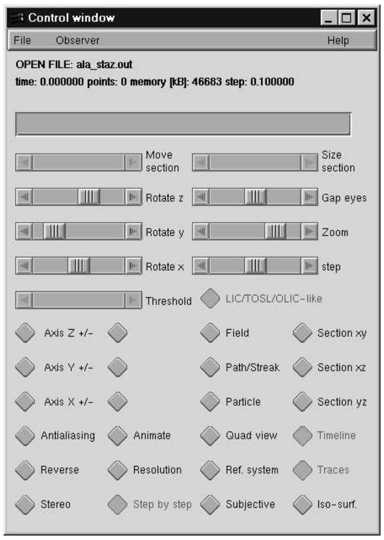

Figure 1: The control window.

celerated hardware which supports the OpenGL. Our program presents the user three windows: a control window (see Fig. 1), a main window (see Fig. 4), and a quad-view window (see Fig. 3). The control window allows the user to change the point of view and the zoom factor, rotate the 3D structure, set the size and the location of the volume slices using a set of slider located in the upper part of the window. The buttons in the lower part allow the user to choose the window to be displayed (main or quad-view window) and a set of other options such as: antialiasing, stereoscopic visualization, reference system visualization, the color of the background (black or white). Moreover, there is a button which allows to start and stop the animation and two other buttons are used to select the kind of particle tracing (pathlines, streaklines and timelines). Three buttons are used to apply a volume slice to the 3D structure; only cut planes along the co-ordinate planes are available. Only the points of the grid and the particles within a volume slice are visualized; in this way the user can inspect more accurately a particular region of the flow field.

The user can set the accuracy of the 3D geometry by a button called *resolution*; in this way, the complexity of the scene can be set according to the computational power available. With the resolution button the user set the granularity of the vector field changing, in this way, the number of grid points to be displayed (in the computation the whole vector field is always considered). Moreover, a button allows the user to choose a visualization of the 3D structure among: points of the grid and velocity vectors, only points of the grid, only velocity vectors, and neither points of the grid nor velocity vectors. A menu bar in the upper part of the control window allows the user to choose other options. In order to enhance the comprehension of the 3D structure the lengths of the velocity vectors are set according to the values of the velocity field in points of the grid; moreover, different scales of color are used for different velocities (lighter colors correspond to higher velocities). Using the menu *Observer*, located on the menu bar, the user can set: the point of view, the look-at point, and the time step of the animation. In order to allow the user future inspections of the 3D data, the simulation can be saved in VRML format: either only the particle traces or the particle traces plus the 3D structure can be saved using the options located on the menu *File*. The number and the location of the seed points is specified by an initialization file without any limitation to the number of insertion points. In the upper part of the control window, the user can find the indication of: the time spent, the memory allocated, the computed points, and the value of the time step.

In the main window the user can see a large 3D visualization of the structure; moreover, for a more accurate inspection, the user can set the stereo modality. We adopt a simple and effective solution for shutter glasses based on an interlaced visualization. The monitor has to be refreshed in interlaced mode using this kind of devices; when the even lines are drawn one view is blocked and when the odd lines are displayed is the other view to be inhibited. A simple and fast way to build stereo pairs is to divide the viewport in two equal parts long the y-axis and to render the images of the two eyes separately into the two viewports. At this point, the final stereo image is composed by picking alternately the rows by the two viewports; the only drawback of this approach is that the vertical resolution is halved but the cost of this kind of device is very low (one hundred dollars) and, hence, well suited for a large-scale employment. In order to obtain an enhanced feeling of immersion, the user can control the interocular gap by a slider located on the control window. No accelerated hardware is required for the stereoscopic visualization; moreover, this solution is platform-independent. In the quad-view window the user see the 3D structure and the three projections along the co-ordinate axis of the particle traces.

4.1 TEXTURE-BASED APPROACHES

When the user inspects the vector field by using volume slices three texture-based visualization techniques can be employed: LIC [Cabra93a], an approach based on sparse textures (OLIC-like) [Wegen97a] and a new method we called TOSL (Thick Oriented Streamline).

TOSL attempts to overcome the drawback of the LIC algorithm which does not allow to identify the flow orientation and the drawback of the techniques based on sparse textures which can loss information. TOSL computes, for each grid cell of the vector field, the local streamline (of a length fixed by the user); at this point, instead of performing a 1D convolution a starting color is chosen in a pseudo-random way and it is assigned to the pixel of the output texture corresponding to the beginning of the stremline under test. The color is varied along the streamline increasing its value according to the magnitude of the vector field in the cell grid considered (each grid cell corresponds to an output pixel). In the event two stremlines intersect, the color of the pixel in the output texture corresponding to the intersection point is computed once when the first streamline is considered.

In order to obtain streamline which have uniform lenghts the strategy for determining which grid cells have to be considered before is very important. The cell grids could be considered in sequence as in the LIC algorithm but this approach could lead to have the first stremlines longer than the other ones; similarly a random method does not guarantee a satisfactory result. This problem can be solved considering a first set of grid cells placed on the vector field according to a Sobol's distribution; after the remaining cells are considered in sequence; a comparison among the three techniques is show in Fig. 2.

5 EXAMPLES

In the first example, we study the flow field induced by two rectilinear vortex filaments, positioned symmetrically with respect to the z-axis, at $y = 0$ and $x = \pm b/2$, and immersed into a uniform stream, parallel to the z-axis. This model has some resemblance with the flow field past a wing of span b. In Fig. 3 we have the quad-view window displaying the streaklines originating from seed locations positioned on a segment parallel to the x-axis ($y = 0$, z=const, $|x| \geq b/2$). A better understanding of the main features of the vortical flow field is obtained displaying the timelines of several particles seeded on the segment $|x| \leq b$, $y = 0$, z=const (see Fig. 4). The structure of the trailing vortices is clearly depicted by the timelines.

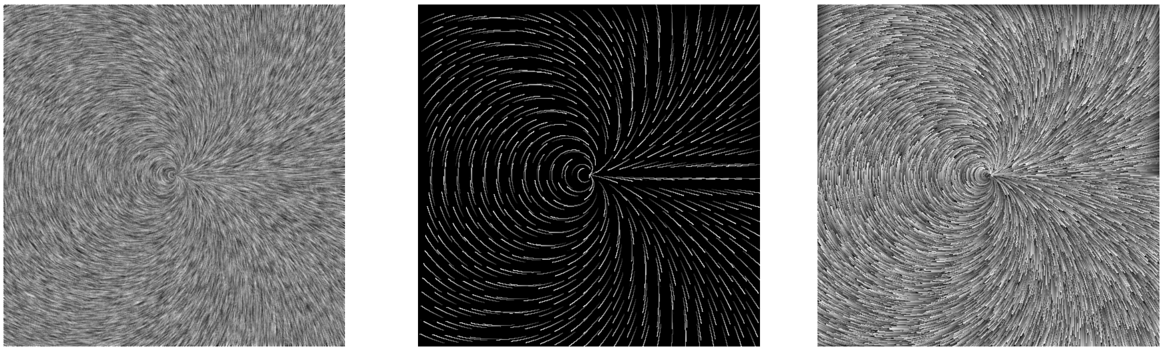

Figure 2: The left image is obtained by LIC, the central image by an approach based on sparse textures and the right image by TOSL.

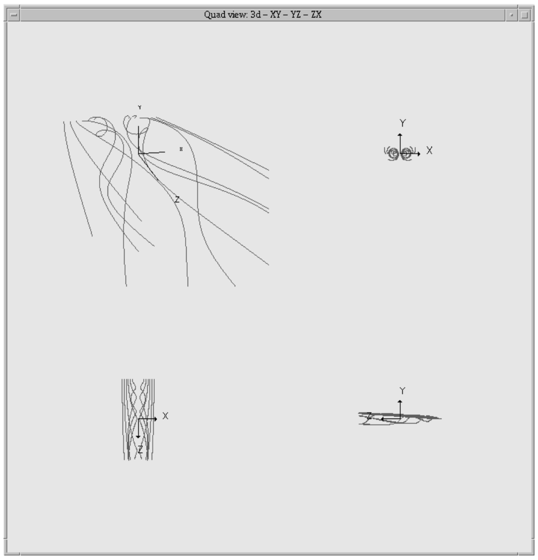

Figure 3: Example 1 - streaklines.

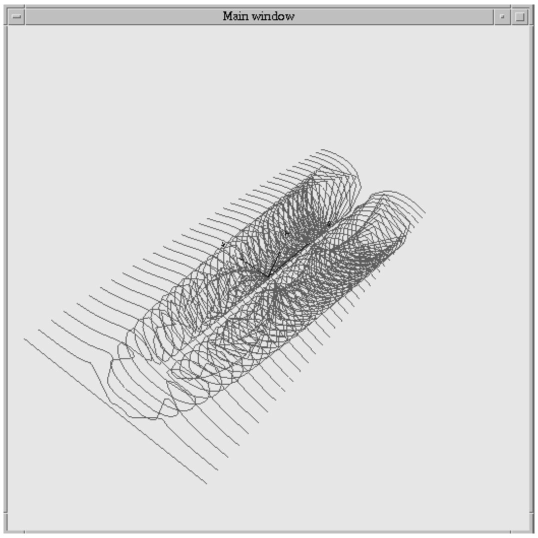

Figure 4: Example 1 - timelines.

In the second example, the two vortex filaments are not kept fixed anymore. By mutual induction they rotate around the z-axis. The flow field is unsteady, moreover the rotation of the vortex filaments introduces an additional complexity into the flow field. In Fig. 5 and Fig. 6, the streaklines and the timelines, obtained in a way similar to the previous example, are displayed. It is evident that a clear description of the flow field may be obtained only with the timelines.

As a last example, we visualize a shear flow, with an unsteady velocity wavy perturbation. Numerical simulations of turbulent flow fields along walls lead to similar results. Fig. 7, where the resulting timelines on a representative section are displayed, shows how the present software may be useful for the visualization of this kind of data. We compared the performance of our software by using a 433 MHz DEC alpha workstation and a 200 MHz Pentium Pro. The workstation was equipped by an accelerated graphics board, threfore, we could compare the performance both in the event of hardware acceleration and in the event of software emulation by the MESA libraries. When the hardware accelartion is used, the geometries considered in the proposed examples can be investigated by the user in real time; the software emulation is about three times slower but the user can however inspect interactively the simulations.

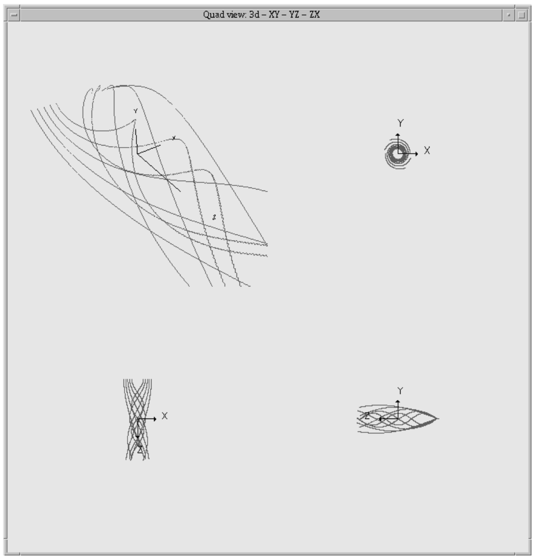

Figure 5: Example 2 - streaklines.

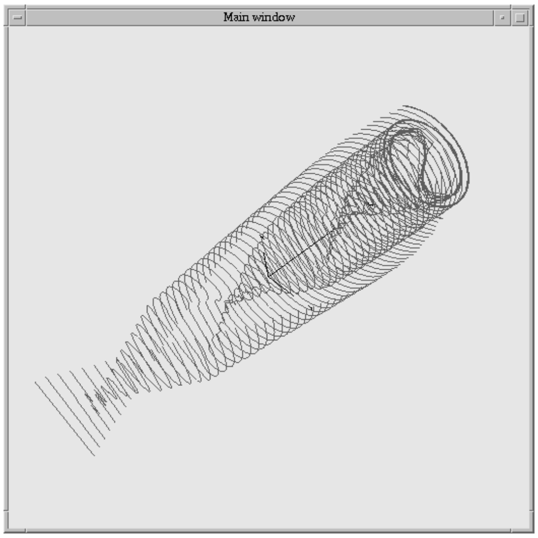

Figure 6: Example 2 - timelines.

The Pentium Pro is not provided of accelarted hardware and is on the average about 2-3 times slower than the workstation. Anyway, also the Personal Computer allows the user a good interactivity in exploring the simulations. For instance, 20 time units of simulation (time step 0.1, 12 insertion points, 47 MB of database and about 250.000 computed points) for obtaining the streaklines of example 1 require about 4 seconds on the workstation and 14 on the PC.

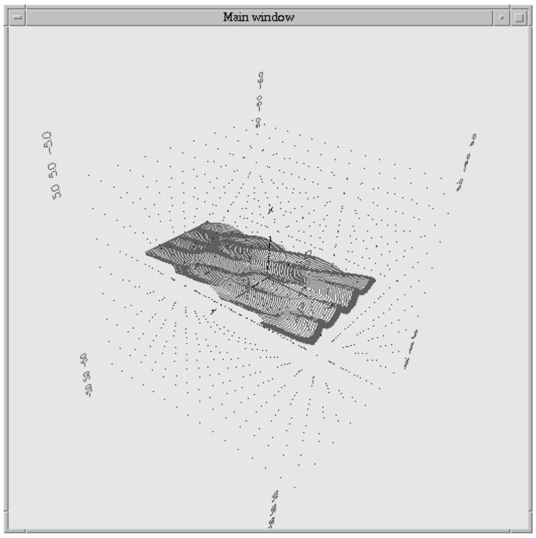

Figure 7: Example 3 - timelines.

6 CONCLUSION

Even if designed and implemented to be used on low-end machines, the proposed software allows the user an accurate inspection both of the 3D structure and of the particle traces. The program supports a stereoscopic visualization device based on interlaced liquid crystal shutter glasses and it allows the user to investigate a specific part of the structure by volume slices or 2D projections; the complexity of the geometry can be controlled by the user which can adapt the work load to the available computational power. Future 3D visualizations can be performed by saving the data in VRML format. Moreover, the program allows the user to visualize the vector field by three texture-based approaches and, in particular, by a new technique called TOSL which overcomes the drawbacks both of the LIC technique and of the methods based on sparse textures.

Future work will be aimed to add new investigation tools such as isosurfaces; moreover, we want to develop a new inspection modality which will allow the user to navigate along the fluid flows as if the user were a particle.

REFERENCES

[Lane96a] Lane, D. A.: *Visualizing Time-Varying Phenomena in Numerical Simulations of Unsteady Flows*, AIAA-96-0048, 34th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 1996.

[Niels97a] Nielson, G. M., Hagen, H. and Müller, H.: *Scientific Visualization*, *IEEE* Computer Society, Los Alamitos, California, 1997.

[Cabra93a] Cabral, B. and Leedom, L. C.: *Imaging Vector Fields Using Line Integral Convolution*, ACM Computer Graphics (Proc. SIGGRAPH'93), Vol. 27, pp. 263-272, 1993.

[Wijk93a] Wijk, J. J. van: *Flow Visualization with Surface Particles*, *IEEE* Computer Graphics & Applications, Vol. 13, No. 4, pp. 18-24, 1993.

[Forse95a] Forsell, L. K. and Cohen, S. D.: *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, pp. 133-141, 1995.

[Wegen97a] Wegenkittl, R. and Gröller, E.: *Animating Flowfields: Rendering of Oriented Line Integral Convolution*, Proceedings of Computer Animation'97, Geneva, Switzerland, 1997.

[Shen98a] Shen, Han-Wei and Kao, D. L.: *A New Line Integral Convolution Algorithm for Visualizing Time-Varying Flow Fields*, *IEEE* Transaction on Visualization and Computer Graphics, Vol. 4, No. 2, pp. 98-108, 1998.

[Inter98a] Interrante, V. and Grosch, C.: *Visualizing 3D Flow*, *IEEE* Computer Graphics & Applications, Vol. 18, No. 4, pp. 49-53, 1998.

[Fuhrm98a] Fuhrmann, A. and Gröller, E.: *Real-Time Techniques for 3D Flow Visualization*, Technical Report, TR-186-2-98-16, Institute of Computer Graphics 186-2, Technical University of Vienna, 1998. Downloadable from: ftp://ftp.cg.tuwien.ac.at/pub/TR/98/

[Bunin85a] Buning, P. and Steger, J.: *Graphics and Flow Visualization in Computational Fluid Dynamics*, AIAA-85-1507, 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, 1985.

[Smith89a] Smith, M., Dalsem, W. van, Dougherty, F., and Buning, P.: *Analysis and Visualization of Complex Unsteady Three-Dimensional Flows*, AIAA-89-0139, 27th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 1989.

[Jespe91a] Jespersen, D. and Levit, C.: *Numerical Simulation of Flow Past a Tapered Cylinder*, AIAA-91-0801, 29th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 1991.

[Haime94a] Haimes, R.: *pV3: A Distributed System for Large-Scale Unsteady CFD Visualization*, AIAA-94-0321, 32nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 1994.

[Neide93a] Neider, J., Davis, T. and Woo, M.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL, release 1*, Addison-Wesley, 1993.