Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Graphics 29 (2005) 17–27

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Virtual Tubelets—efficiently visualizing large amounts of particle trajectories

Marc Schirski[a,*], Torsten Kuhlen[a], Martin Hopp[b], Philipp Adomeit[b], Stefan Pischinger[c], Christian Bischof[a]

[a]*Center for Computing and Communication, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, Germany*
[b]*FEV Motorentechnik GmbH, Neuenhofstr. 181, 52078 Aachen, Germany*
[c]*Institute for Internal Combustion Engines (VKA), RWTH Aachen University, Schinkelstr. 8, 52062 Aachen, Germany*

Abstract

The depiction of particle trajectories is an effective means for the visualization of fluid flows. However, standard visualization techniques suffer from a variety of weaknesses, ranging from ambiguous depth perception for simple line drawings to a high geometrical complexity and decreased interactivity for polygonal tubes. This paper addresses these problems by introducing a novel approach to pathline visualization, which we call *Virtual Tubelets*. It employs billboarding techniques in combination with suitable textures in order to create the illusion of solid tubes, thus efficiently and unambiguously depicting large amounts of particle trajectories at interactive frame rates. By choosing an appropriate orientation for the billboards, certain issues concerning immersive displays with multiple projection screens are resolved, which allows for an unrestricted use in virtual environments as well. Using modern graphics hardware with programmable vertex and pixel pipelines results in an additional speed-up of the rendering process and a further improvement of image quality. This creates a nearly perfect illusion of tubular geometry, including plausible intersections and consistent illumination with the rest of the scene. The efficiency of our approach is proven by comparing rendering speed and visual quality of *Virtual Tubelets* to that of conventional, polygonal tube renderings.

Keywords: Virtual reality; Scientific data visualization; Viewing algorithms; Programmable graphics hardware

1. Introduction

Analyzing the movement of particles through a volume is an intuitive and effective means for understanding the underlying flow. This includes massless particles, which are inserted into the flow as a post-processing measure, as well as the results of dispersed multi-phase flow models [1], which simulate the behavior of, e.g. liquid fuel drops inside a combustion chamber, and which generate additional data like temperature or radius. While displaying the motion of these particles in an animated sequence is a valid method of visualizing their trajectories, it loses expressiveness as soon as the animation is stopped. On the other hand, depicting the whole trajectory of every single particle at once results in a cluttering of the view space due to the sheer amount of information displayed at any time, especially when visualizing particle movement over prolonged time periods. In this case, occlusion becomes a major problem. Furthermore, the informational connection between a particle's location and the corresponding time

*Corresponding author.
E-mail address: schirski@rz.rwth-aachen.de (M. Schirski).

0097-8493/$ - see front matter
doi:10.1016/j.cag.2004.11.004

is lost. We address these problems by displaying particle traces, i.e. the particle position at a given time, supplemented with a tail of user-definable length to display a particle's recent history. This allows for a better understanding of a particle's behavior in time, while minimizing screen clutter.

Common issues of line-based trajectory visualizations include depth perception issues and ambiguities concerning the depth order. A very common workaround is the depiction of particle traces as geometrical tubes, which in turn increase the geometrical complexity of the scene. This results in a significantly higher load on the graphics system and decreased frame rates, thus reducing the responsiveness and interactivity of the visualization. Actually, this is one of the major problems in employing this visualization method in Virtual Reality (VR) applications, because head-tracking and the ubiquitous need for interactivity requires the frame rate to stay above a certain threshold at any time. Furthermore, substituting simple lines by complex geometry reduces the flexibility of the visualization, because every change of visualization parameters (including the currently displayed time step) requires a recreation of this geometry. This can be reduced somewhat by precomputing tubular geometry for every single time step, which in turn increases memory consumption significantly.

In this paper, we show a way of substituting complex tubular geometry by cylindrical billboards in order to depict particle trajectories. In their most basic form, they consists of user-aligned quadrilaterals with a suitable texture map to create the illusion of slightly self-illuminated, rounded geometry, while still maintaining high frame rates (see Fig. 1). Perception problems caused by abrupt endings of these virtual tubes are avoided by an explicit capping. This approach allows for a high responsiveness and interactivity of the visualization system in combination with a high visual quality and clarity. In addition, the absence of a complex geometrical representation in the form of polygonal tubes leads to a high flexibility regarding view parameter changes and low memory consumption. Using modern graphics hardware allows for an even more convincing simulation by incorporating per-pixel bump mapping and depth replacement, thus creating a nearly perfect illusion, even in combination with conventional geometry. Of course, all principles shown in this paper can be applied to an efficient rendering of conventional streamlines as well.

The remainder of this paper is structured as follows: Section 2 gives an overview of related work and Section 3 deals with a basic form of *Virtual Tubelets* and its implementation. The application of *Virtual Tubelets* in virtual environments is discussed in Section 4. In

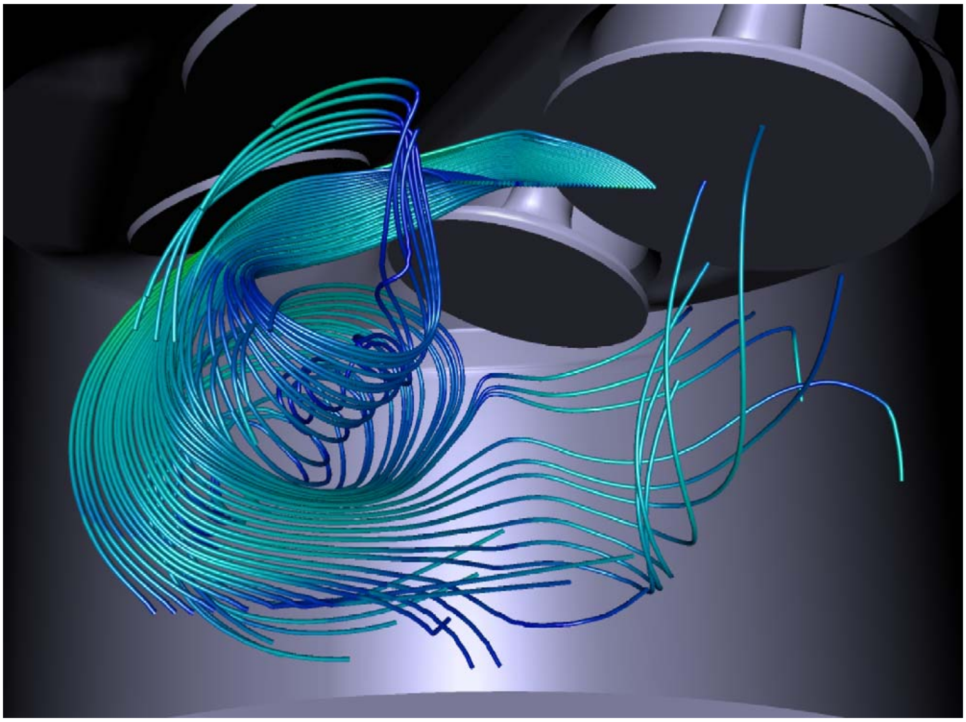

Fig. 1. Visualizing pathlines via *Virtual Tubelets*; specular and diffuse illumination are computed in a fragment program.