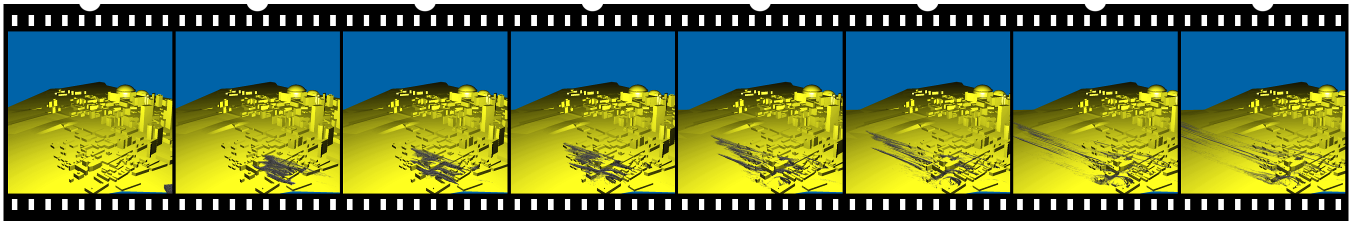
# Visualization Contest 2005 - All you need is ... - Particles!

Jens Schneider, Polina Kondratieva, Jens Krüger, Rüdiger Westermann*
Computer Graphics and Visualization Group, Technische Universität München



*The New Orleans data set visualized by means of particle tracing. The particle source is located at Poydras Street Wharf, Riverwalk Marketplace. With 735 x 125 x 50ft it has about the same size as the freighter Bright Field, though it is rectangular. Particles are emitted in a single burst to simulate the impact, and are then advected with the wind field. Note that while some particles are transported away rather quickly, others are captured in vortices and stay for a long time in the same region (most prominent in the foreground). This visualization was done using 64K semi-transparent, depth-sorted, and textured splats and runs at 25 fps. Without depth sorting, we achieve approximately 171 fps.*

## 1 Introduction

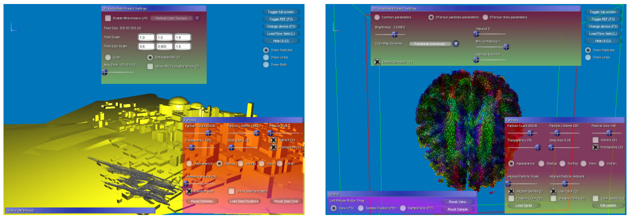


Figure 1: *Screenshots of the particle engine, showing only a small fraction of the plethora of available settings. Left: Particle and data set menu, right: Particle and DTI menu.*

We applied our particle engine as described in [Krüger et al. 2005] to efficiently and effectively render both of the data sets. This particle engine is custom-taylored for interactive visualization of steady 3D flow fields, such as the New Orleans wind field. To achieve high frame rates for a large amount of particles, features of recent graphics accelerators to advect particles in the graphics processing unit (GPU) are exploited. Particle positions and attributes, as well as the velocity field are kept in graphics memory, removing the need to transfer any data across the graphics bus for rendering. This allows for the rendering of millions of particles at interactive rates.

The user can choose from a toolbox of visualization methods, such as plain particles, oriented and textured splats, shaded lines, stream balls, or stream ribbons, and can augment the result even further by means of volume rendering or the display of triangle meshes. This enables the user to map different rendering methods to different modalities of the data at interactive framerates. In figure 1 we show screenshots of the application, demonstrating the multitude of rendering modes available. Even more, since the entire application is based on DirectX' Techniques, it is very easily extended by new modes.

All timings were done on a Pentium IV with 2.4GHz and 1GB of RAM, equipped with an *n*Vidia GeForce 6800 Ultra.

## 2 New Orleans Windfield

### 2.1 Pre-Processing

The New Orleans data set came sampled on an unstructured tetrahedral grid. The area around the houses in the middle are resolved very finely, while the rest of the data set is sampled only coarsely, and is presumably only there to provide enough space to avoid reflections at the simulation boundaries. Since the particle engine operates on regular carthesic or rectilinear grids for performance reasons, we had to convert the data to either one of the two formats. Taking a close inspection of the distribution of the tetrahedra volumes across the city we realized that a regular cartesic grid would be sufficient, and would result in slightly better rendering performance. Consequently we resampled the area around the city to a $512^2$x64 data set, containing for each voxels the 3D momentum and the scalar energy as 32bit float values. The data set also contained a density entry, which is 1.0 across the entire domain. For resampling the data we rasterized (on CPU) the tetahedra into the volume, using spherical interpolation of the momentum vectors and linear interpolation of the scalar energy entries. For both quantities we achieved a scalar PSNR of approximately 50dB. The data set contains some degenerated tetrahedra which we removed, and some holes which we fixed by simple region growing. The city mesh was cut out of the boundary triangles and was reduced to about 26K triangles using standard meshing operations based on error quadrics.

### 2.2 Visualization

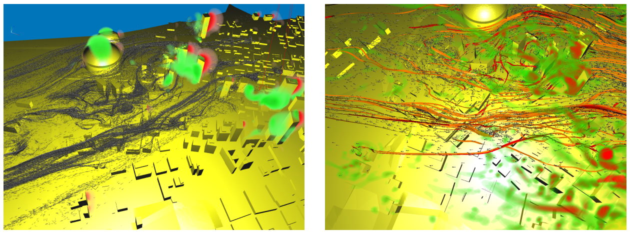


Figure 2: *Flowfield close to the Louisiana Superdome. Left: oriented particles and the energy volume are combined, highlighting regions with high kinetic energy in green while those with high pressure are colored red (41 fps) Right: the energy volume was replaced by a $\lambda_2$ volume. Vortex cores are emphasized using a simple red-to-green transfer function. This image also demonstrates the use of stream-ribbons. (22 fps)*

The method that gave the best impression of the data was to

*{schneidj, kondrati, kruegeje, westerma}@in.tum.de

trace shaped, momentum aligned particles through the wind field, and augmenting them with a volume rendering of the energy field (c.f. 2 left). Throughout the visualizations made for this contest, all primitives requiring advection (lines, particles etc.) where traced using an embedded RK32. The energy field is in a signed format, and most likely contains the difference between static (i.e. pressure) and dynamic (i.e. kinetic) energy. Applying a transfer function that highlights regions of positive energy green and regions of negative energy in red gives precise information of endangered regions in case of air-borne contaminants: In red regions pressure is high, such that these regions tend to saturate with the contaminant, while green regions tend to have high contamination flux. However, our experiments with the data lead us to the suspicion that some of the green regions, namely behind larger buildings, are fairly safe, since they are comparable to an eye of a whirlwind, being enclosed by vortical flow.

To understand the vortices in the present data set, we computed a $\lambda_2$ volume similar to [Stegmaier and Ertl 2004], but on the CPU. The name stems from the fact that isosurfaces of small and negative second eigenvalue of the real, symmetric tensor $\Omega^2 + S^2$ enclose vortex tubes [Jeong and Hussain 1995]. $\Omega$ and $S$ refer to the antisymmetric and symmetric parts of the Jacobian of the underlying velocity field. The result are tube-like structures behind various buildings that clearly communicate the turbulent behaviour of the windfield (c.f. 2 right). Again, various rendering modes where combined.

# 3 DT-MRI Brain

## 3.1 Pre-Processing

Basically the reconstruction of DT-MRI data sets is straight forward. DT-MRI data sets consist out of a reference scan and $N$ scans obtained under a gradient field. As long as all of the $N$ gradient directions and the B-value of the field are known, the reconstruction follows standard literature [Hasan et al. 2001; Westin et al. 2002]. This data set, however, was obtained using a Philips device, which stores valuable information not in standard, but in proprietary DICOM tags. Consequently the most demanding part was to figure out the number of gradients, the strength of the field, and the ordering of slices. Since we did not have a Philips-compliant reader, we resorted to the freely available tool dicom2 [Barré n. d.] to extract the pixel matrix and the header from each slice. Though these proprietary tags are documented [Philips 2002], they had to be extracted by hand. The ordering of the slices was inferred from the “Image Position (Patient)” tag, resulting in 17 volumes with each 24 slices. The B-value can be found in the proprietary “Diffusion B-Value” tag. Since it is 0 for the reference image, those slices are easily identified. The gradient directions are stored in a - at least to our knowledge - undocumented way. Retrieving all unknown tags containing a single float and inspecting them yielded plausible gradient directions in three consecutive tags with numbers 0x2005:0x10B0 thru 0x2005:0x10B2. There are exactly 16 non-standard directions among all slices (Philips otherwise uses icosahedral directions), one of which is zero. This partitions the set of remaining slices into 15 valid gradient volumes and one zero-gradient volume. A first inspection of the reconstructed data shows that it is very noisy and that some information seems to be destroyed beyond recovery. To reduce the noise we applied to each volume both a thresholding based on the reference image, and a $3^3$ median filter. To segment the interior of the brain, we also used a manual segmentation of the reference volume in order to remove noise outside the head. Still, the low resolution of only 24 slices makes visualization of this data set a difficult task. We would like to refer to our application paper [Kondratieva et al. 2005] in this very conference for the visualization of high-resolution data sets.

## 3.2 Visualization

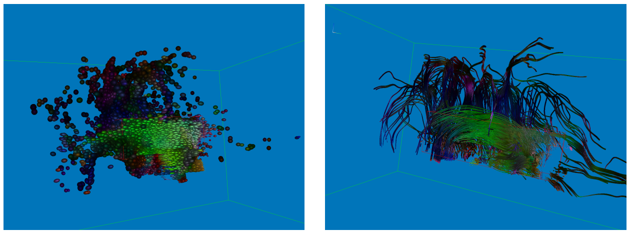


Figure 3: *The DTI brain data set. Left: Visualization using diffusion oriented, ellipsoidal splats. The splats are deformed according to the anisotropy of the tensor and removed in regions where the diffusion tensor is ambiguous. (15 fps) Right: A similar region is visualized using stream tubes. Stream tubes better convey the impression of fibers, but lack to show the dynamic behaviour in animations. (5 fps) For both images the fractional anisotropy (FA) color model was chosen.*

We follow the visualization modes presented in the aforementioned paper [Kondratieva et al. 2005]. We store in two float textures the six degrees of freedom of the symmetric diffusion tensor, as well as a confidence value. The confidence value is mainly used to mask out noisy areas. Then, for each particle position, the neighboring diffusion tensors are interpolated, and the particles are advected using either the first eigenvector or deflection [Kindlmann 2003]. To emphasize the anisotropy of the tensor, and to show the fiber tracts, the user can select freely from different rendering primitives such as shaded lines or diffusion-aligned, deforming ellipsoidal splats (c.f. 3). The visualization can be augmented with a volume rendering of the reference image to provide addtitional context. Also, the user can select among different color schemes from standard literature [Kindlmann 2003], such as $C_p, C_l, C_s$, or $FA$-based. Though the data set is far from optimum quality, some prominent structures such as the superior longitudinal fasciculus (green) can be distinguished.

# 4 Additional Material

http://wwwcg.in.tum.de/vis-contest

# References

BARRÉ, S. Dicom2. http://www.barre.nom.fr/medical/dicom2/.

HASAN, K., PARKER, D., AND ALEXANDER, A. 2001. Comparison of gradient encoding schemes for diffusion-tensor MRI. *Journal on Magnetic Resonance Imaging 13*, 769–780.

JEONG, J., AND HUSSAIN, F. 1995. On the identification of a vortex. *Journal on Fluid Mechanics 285*, 69–94.

KINDLMANN, G. 2003. *DTI Visualization and Analysis of Diffusion Tensor Fields*. PhD thesis, University of Utah.

KONDRATIEVA, P., KRÜGER, J., AND WESTERMANN, R. 2005. The application of gpu particle tracing to diffusion tensor field visualization. In *Proceedings IEEE Visualization 2005*.

KRÜGER, J., KIPFER, P., KONDRATIEVA, P., AND WESTERMANN, R. 2005. A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics 11*, 5 (9).

PHILIPS. 2002. DICOM Conformance Statement. Gyroscan Intera R 7.5.1, R 8.1 and R 9.1. Tech. Rep. 4522 131 88661, Philips Medical Systems, May.

STEGMAIER, S., AND ERTL, T. 2004. A Graphics Hardware-based Vortex Detection and Visualization System. In *Proceedings of IEEE Visualization '04*, 195–202.

WESTIN, C., MAIER, S., MAMATA, H., NABAVI, A., JOLESZ, F., AND KIKINIS, R. 2002. Processing and visualization for diffusion tensor mri. *Medical Image Analysis 6*, 2 (June), 93–108.