

In vivo interactive visualization of four-dimensional blood flow patterns

Realtime assessment of volumetric phase contrast MRI

Bernhard Kainz · Ursula Reiter · Gert Reiter ·
Dieter Schmalstieg

Published online: 3 March 2009
© Springer-Verlag 2009

Abstract In this paper we give an overview over a series of experiments to visualize and measure flow fields in the human vascular system with respect to their diagnostic capabilities. The experiments utilize a selection of GPU-based sparse and dense flow visualization algorithms to show the diagnostic opportunities for volumetric cardiovascular phase contrast magnetic resonance imaging data sets. Besides classical hardware accelerated particle and line-based approaches, an extensible tablet-based visualization, a four-dimensional volumetric line integral convolution and a new two-dimensional cutting plane tool for three-dimensional velocity data sets have been implemented. To evaluate the results, several hearts of human subjects have been investigated and a flow phantom was built to artificially simulate distinctive flow features. Our results demonstrate that we are able to provide an interactive tool for cardiovascular

diagnostics with complementary hardware accelerated visualizations.

Keywords Flow visualization · Cardiovascular diagnostics · Phase contrast magnetic resonance imaging (PC-MRI)

1 Introduction

Assessment of blood flow properties is crucial in the understanding and diagnosis of many cardiovascular diseases. The Magnetic Resonance (MR) through-plane phase contrast method provides useful information from flow through cross sections or velocities in preferred directions. However, its usefulness in situations involving complex fluid dynamics—as for example in the cardiac chambers—is limited, because the main directions of flow are neither known nor constant in time. Conceptually, the easiest way to acquire three-dimensional blood flow data is to measure both through-plane and in-plane velocity components via phase contrast sequences. Velocity vectors are determined on each imaging plane: In the case of the combined through-plane and in-plane measurement for each pixel.

Our motivating application is the accurate non-invasive diagnosis of pathological changes in the human cardiovascular system. Usually such diseases are characterized by an elevation in pressure or the change of common flow patterns. This is quite often invasively diagnosed via catheterization, which is a procedure with mortality risk. Consequently it is desirable to have non-invasive diagnostic methods, which are as significant as invasive ones.

Pathologically altered pressures in the vascular system lead to changes in blood flow patterns. An imaging method

Electronic supplementary material The online version of this article (<http://dx.doi.org/10.1007/s00371-009-0315-7>) contains supplementary material, which is available to authorized users.

B. Kainz (✉) · D. Schmalstieg
Institute for Computer Graphics and Vision, Graz University
of Technology, Inffeldgasse 16a, Graz, Austria
e-mail: kainz@icg.tugraz.at

D. Schmalstieg
e-mail: schmalstieg@icg.tugraz.at

U. Reiter
Graz University Clinic of Radiology, Auenbruggerplatz 9,
8010 Graz, Austria
e-mail: ursula.reiter@klinikum-gaz.at

G. Reiter
Siemens Healthcare Austria, Straßgangerstraße 315, 8010 Graz,
Austria
e-mail: gert.reiter@siemens.com

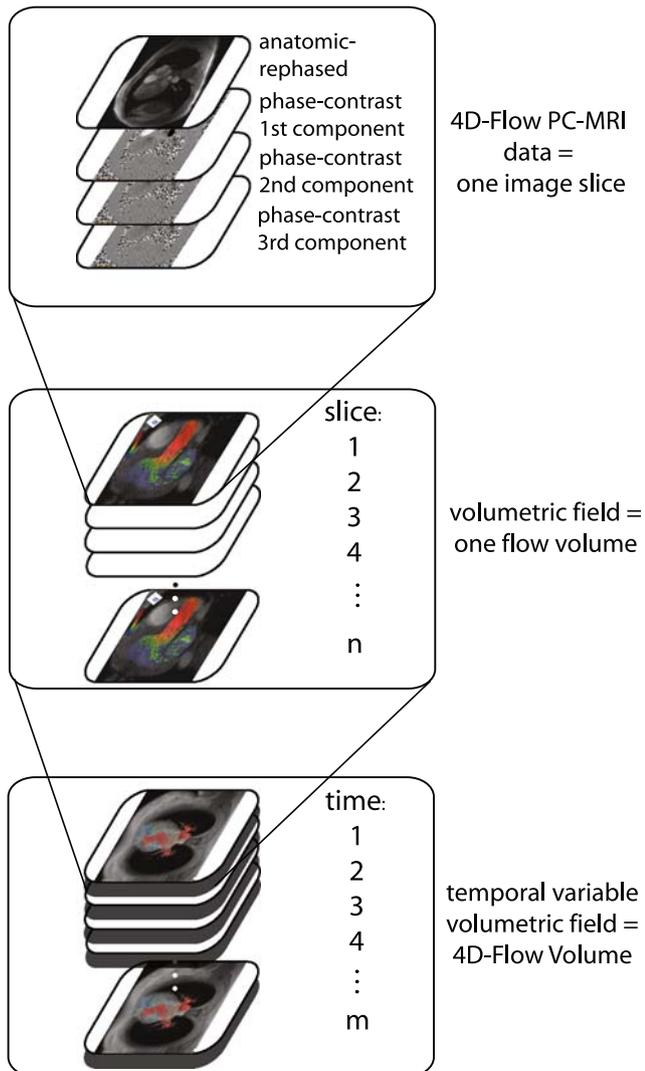


Fig. 1 We use a volumetric time-dependent velocity-encoded data set. The phase-contrast velocity encoded images contain one spacial direction component each. This part describes one slice for one certain volume. One volume consists of several of these slices, so the number of recorded images for one volume is given by the four images which are required for one slice, times the number of slices itself. Since we handle four-dimensional data, there is one volume per time step. Consequently the final number of raw images is $\#images = 4 \cdot \#slices \cdot \#timesteps$

for an accurate measurement of the blood flow is therefore desirable. This imaging method is called phase contrast magnetic resonance imaging (PC-MRI) [4, 17, 21]. The vast amount of data and its complex composition requires the development of tools and methods for a clear representation aimed at physicians who may evaluate possible vascular diseases or understand the human cardiovascular system in a better way. The four-dimensional composition of our investigated data sets is outlined in Fig. 1. High-performance real-time GPU accelerated visualization techniques help to investigate the resulting four-dimensional data sets in this work.

1.1 Contributions

This work concentrates on *point-based*, *sparse* and *dense flow representations* as defined by Weiskopf [27]. We implemented different types of hardware accelerated direct particle-based visualizations and particle trajectory-based line visualizations as well as line integral convolution (LIC) techniques with volume raycasting. It is possible to use all these techniques simultaneously without a loss of interactivity. Additionally we provide several cutting plane visualizations and improvements of basic streamline approaches. In particular we developed a novel approach to reintegrate 3D information in 2D cutting planes by using velocity directions as plane normal vectors. Furthermore we provide an enhanced pathline approach to get the connection to time dependent anatomic structures.

1.2 Paper outline

In Sect. 2 we give an overview of related work on PC-MRI data preparation and appropriate flow visualization techniques. Section 3 describes the underlying MRI sequence which we use for our experiments. Results are shown in Sect. 4.

2 Related work

In the area of blood flow analysis Castro, Cebra and Frangi have delivered major contributions. They showed, that flow analysis is already an important diagnostic tool for cerebral aneurysms [3]. Several publications have demonstrated a correlation between vascular diseases and hemodynamics [10, 19, 20]. Primarily systolic flow was studied in the last decade, since the functional efficiency is the crucial factor for the pumping function of the heart [5, 24, 26].

Bringing together the diagnostic possibilities of flow imaging and the research which was done on flow visualization techniques is one of the main objectives of this work. Within the last decade a vast number of flow visualization methods were developed. Wigstroem et al. [28] for example described flow trajectories displaying 3D particle traces within anatomical projected morphologic 2D slices. They found that this type of visualization creates an intuitive display of complex intracardiac flow patterns, but were not able to describe more than a limited number of trajectories concurrently.

Hardware accelerated visualization algorithms have been introduced since the upcoming of general purpose processing on GPU. Krueger et al. [9], Kolb et al. [8] and Latta [12] have demonstrated the possibilities of particle systems calculated on graphics hardware. Knowing the path of a particle leads directly to more challenging visualizations of the

trajectories. Implemented on graphics hardware, they can be used as a powerful tool for diagnosis. However, modern graphics hardware allows more complex numerical integration methods, which differ in their accuracy from former approaches and can be executed even faster than simple calculation schemes on previous hardware. Basic flow visualization approaches can be distinguished between two main representations: *sparse* flow field *representations* and *dense* flow field *representations*.

Sparse representations use mostly particle tracing methods and obvious array/glyph and pseudo-color visualizations. The spatial domain is not densely covered. Among these are single particle effects, pathlines, streamlines, timelines and streaklines. A streamline is defined as a line that is the tangent to the velocity vector of the flow. Thus this line represents the curve a particle takes when it is injected into a static flow field. The trajectory of a particle in a temporal changing field is called pathline. The graphical representation of streamlines or pathlines can either be a “flat” line segment, a streamribbon or a streamtube. For line segments, a simple series of lines are drawn. To represent streamribbons, a polygonal mesh is created by bridging two adjacent field lines. Good results can be obtained for streamribbons, streamsurfaces and streamtubes as long as the adjacent field lines converge [7].

A couple of problems arise with the visual presentation of the results. These include perceptual problems due to projection, occlusion and cluttering of the images. One possibility to improve the spatial perception is to add illumination to the scene (for example Phong-(line)-Shading instead of flat shading, see [1]). Displaying too many field lines will result in a cluttered image, so the visualization is limited to a number of seed points. According to [27] two seed placement strategies exist: interactive or automatic placement. Helgeland [7] describes different methods for automatic seed point placement that include a completely random approach and one that uses a preprocessing step with a simple threshold filter to find seed points. Another method is mentioned by Weiskopf [27] who uses a uniform distribution of streamlines which is likely not to miss important features. However, to analyze special flow patterns in defined regions, a manual seeding strategy is still accepted.

Dense representations typically use texture-based techniques and are therefore sometimes called texture-based techniques. Again a distinction is drawn between sparse and dense representation. However, this differentiation should not be taken too strictly because both use particle tracing to form the final image [27]. The two most known approaches that fall into this category are spot noise [29] and line integral convolution (LIC) [2]. Van Wijk’s approach [29] generates spots on a spatial domain and generates a streak in flow direction, which represents the particle advection at this point [27]. It is capable of reflecting velocity magnitudes by

the amount of smeared texture [11]. By selecting the number of spots, a tradeoff between computational speed and image quality can be made. The enhanced Spot Noise uses bent spot primitives to improve rendering of areas with high local curvature but implies higher costs of computation [11, 14].

The original two-dimensional LIC algorithm was developed in 1993 by Cabral et al. [2] and has become a widely used technique for dense streamline representation. It takes a random white noise texture and a vector field as input. LIC then does a convolution with a kernel filter along the streamlines which causes voxel intensities to be highly correlated along the field lines but independent in perpendicular directions [2, 27].

Both dense techniques are not interactively usable for large images without any optimizations [16] or hardware acceleration [6, 23] of the algorithms. Furthermore, higher dimensional approaches require an proper visualization of the volume.

3 Method

3.1 Definition of the MRI sequence

We use a phase-contrast magnetic resonance flow quantification method (phase-contrast magnetic resonance angiography—PC-MRA) to acquire feasible flow data similar to [28]. A phase-contrast cardiac sequence is used to acquire in vivo data sets.

Cine phase-contrast MRI measurements are performed with a fast low-angle shot (FLASH) gradient-echo sequence and with electrocardiographic (ECG) retrospective triggering. The data is acquired from a human subject and a flow phantom using a 1.5 Tesla scanner. The measurement parameters for the data set presented here are as follows:

- $T_{\text{Repetition}} (T_R) = 89 \text{ ms}$
- $T_{\text{Echo}} (T_E) = 4.1 \text{ ms}$
- Slice thickness = 3–6 mm
- Flip angle = 15°
- $FoV_{\text{read}} = 340 \text{ mm}$
- $FoV_{\text{phase}} = 81.3\%$
- Matrix size = 252×385
- Base resolution = 192 pixel
- Phase resolution = 73%

Between 10 and 30 slices are recorded consecutively at intervals of 4–5 mm for one volume with 20 calculated phases. The complete acquisition time using a GRAPPA sequence is approximately 25 minutes. Velocity encoding is chosen between 90 and 400 cm/s for each direction. The resulting interpolated matrix size is 252×385 [pixel] for each slice in time. Considering 12 Bytes per recorded pixel one slice image is about 1.1 MB. Adding the velocity encoded

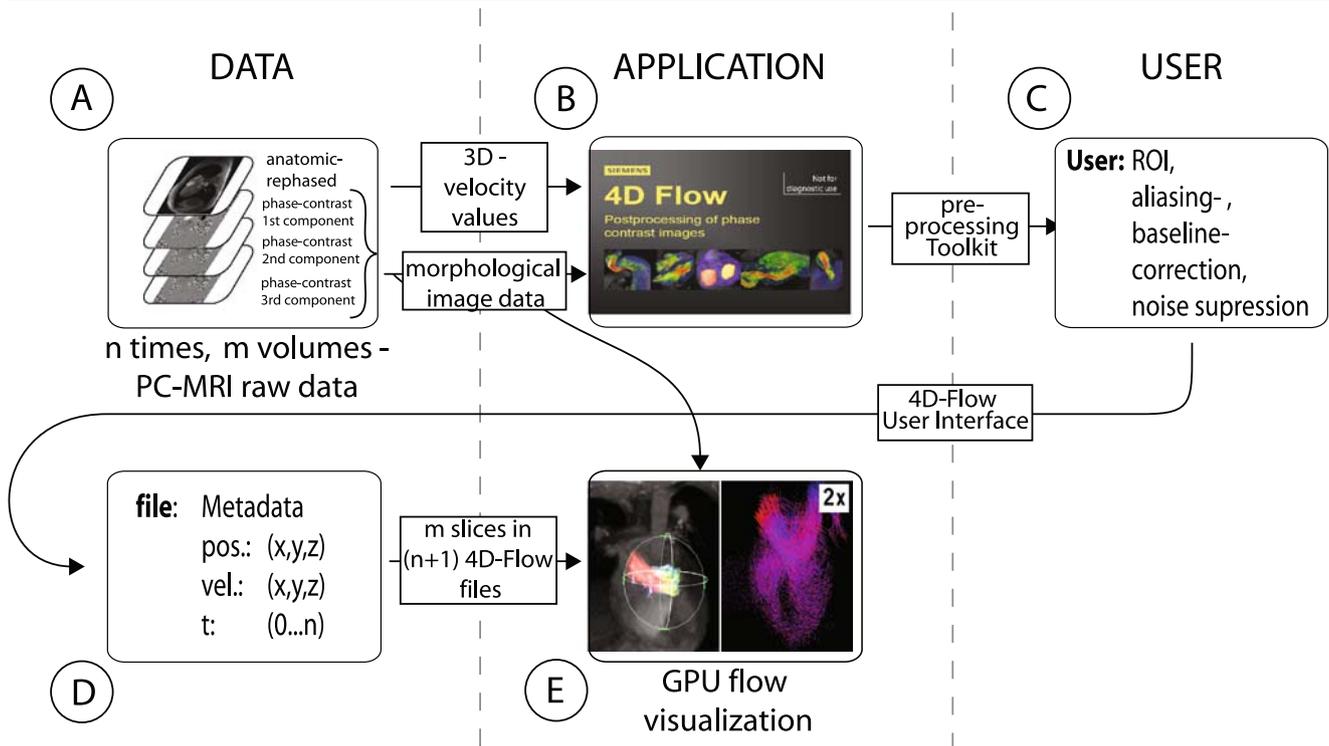


Fig. 2 This workflow diagram describes the traversal of the measured raw data. Data acquisition is marked with (A). The next step is data pre-processing (B). With the preprocessing, the user has to perform some data improvement (C). The processed data is stored in files (D). These

files are used in combination with the anatomic raw images from (A) by our flow visualization framework (E). (A) and (D) belong to the data layer, (B) and (E) to the application layer and (C) needs user interaction. The arrows additionally indicate the course of the data flow

images and an average of 20 slices per time step and an average temporal resolution of 10, the size for one data set adds up to about 888 MB. Defining a region of interest can reduce this size to one third.

Preprocessing of the data sets is outlined in Fig. 2. We use a preprocessing toolkit [21] for aliasing corrections, baseline corrections and noise corrections. The resulting corrected data is used for our core visualization module.

3.2 GPU accelerated sparse visualization algorithms

A particle system with predefined velocity fields can be implemented on GPU hardware with two double buffered textures for position and attribute storage and a 3D Texture for the current flow volume. An additional predefined and constant texture is used to store the random initial positions within $[-1 + \epsilon, 1 - \epsilon]$, where ϵ defines the initial scatter of the particles around a given coordinate point. Figure 3 outlines this concept.

Three subsequent shaders are used to provide an iterative traversal of the algorithm. The first it looks up the current position of a particle in a velocity field-texture and updates its position depending on the found velocity value. The result is stored in a temporally texture. The color map can

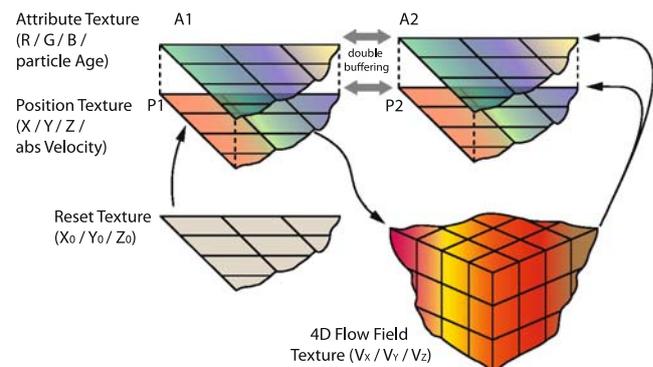


Fig. 3 A double buffered *Position Texture* stores the current position of each particle and an *Attribute Texture* stores the color assigned to a certain velocity or the particle. The *Reset Texture* is used if no velocity value has been found or if the particle left the volume or exceeded its maximum lifetime. The initial positions, which are stored in this texture, are predefined randomly, but they can also be arranged regularly, so that streaklines will result. The *Field Texture* stores the current flow volume for texture lookup and can be changed during rendering in case of four-dimensional volumes

be defined by an intuitive editor as it is shown in Fig. 4. The second shader calculates a color for the particles with a given color-map and the particle's speeds and stores the color in another temporally texture. However, speed is not

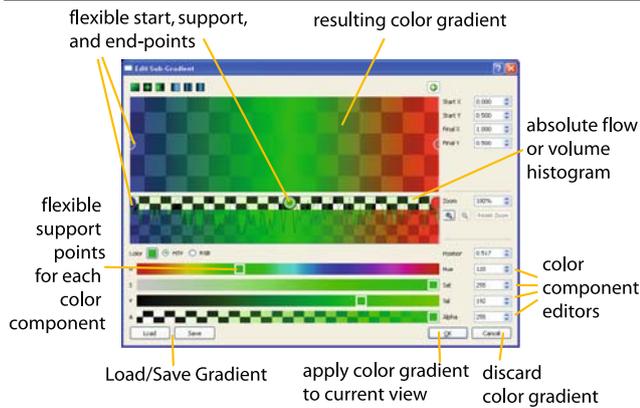


Fig. 4 Mapping the amplitude of a characteristic flow value like velocity, bending energy or relative pressure to a color gradient. With a color gradient editor, it is possible to define arbitrary supporting points for each color-component’s mapping function and an optional transparency function. By applying a new gradient to a view, this gradient will serve as lookup table for a defined parameter. The transparency (alpha) value of the gradient can be used to fade out currently uninteresting parts of the flow, e.g. slow velocities

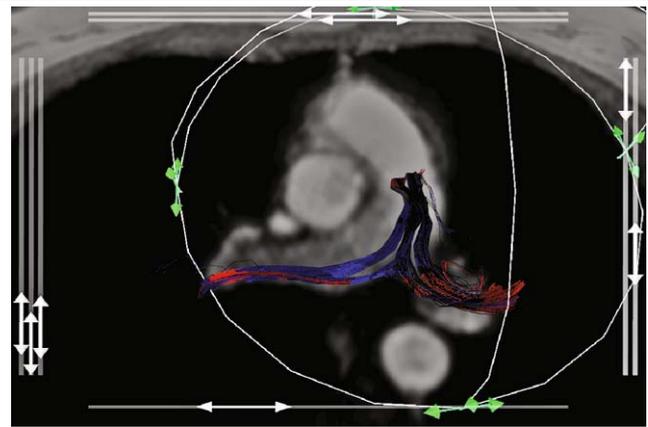


Fig. 5 The bifurcation of the pulmonary artery supported with path-line flow visualization. The parts which refer to the time of the underlying anatomical information are colored *red*. The seed points of the lines can be manipulated with a so-called *dragger* in case of a desktop setup. The dragger is rendered in this example as a white spherical structure around the investigated area

the only feature which can be mapped to a certain color. Due the modular shader design, the mapping can be exchanged to, for example, relative pressure measures or centrifugal force during runtime. The third shader sets the number of available particles to the desired positions and displaces them accordingly to the concurrently performed calculations of the trajectories.

The trajectories itself are numerical integrated by a Runge–Kutta approach of fourth order. Compared to an explicit Euler method, it produces more accurate results with less sampled supporting points of the desired trajectory. This technique is favored for all numerical integrations.

Streamlines show the trajectories of a particle influenced by a static velocity field. Consequently, these lines are only valid for one separate temporal volume and will completely change when changing to another field in time. However, local structures of the flow may be explored very well by these lines. In contrast to flowing particles lines require all sampling points for visualization. The buffer strategy from above was modified so that each row in the textures stores the sampling points of the trajectories for one time step. Not all lines are of the same size because they may be interrupted by other lines or end at the volumes borders. Therefore, a magic number is stored in the alpha channel of the *attribute texture* to indicate the start of a new line. The line evaluation is stopped and therefore interrupted if other pre-calculated line segments have already been marked as drawn. This avoids a cluttering between crossing line segments.

Calculation of pathlines is built upon the calculation of streamlines. This line type has to consider the fourth dimension as well, since a pathline represents a particle’s trajectory over time. The simplest way to generate a pathline is to assemble it with parts of streamlines from each volume. This

method is error-prone since, there is no or only interpolated additional information between the time steps. However, this method is efficient to implement by changing the volumes over time during the calculation of streamlines. Assuming that a given four-dimensional data set consists of t volumes and the position buffer texture is of the size $n \times m$, the volume number has to be incremented each $\text{floor}(\frac{m}{t})$ frame during the line trace. Since these lines are now constant when the user for example changes the image volume, an additional indicator is required for convenience. We found that a simple but novel approach can bypass the problems which user have when they need to assign certain parts of a pathline to the additionally shown anatomical information. Coloring parts of the lines with a signal color shows the user in which volume which part of the line was generated. We use colors of high contrast. Figure 5 shows an example of our pathline color indication method.

Based on pathlines and streamlines, a more realistic visualization, so-called streamtubes are possible. Generating a mesh of vertices around a line would result in a vast amount of vertex calculations and consequently in a very low performance. To handle this problem, an impostor rendering technique was used. This method allows to render reasonably realistic tubes with only twice as many vertices as used with simple line visualizations. Instead of a line strip a quad strip¹ is stored in a display list and then subsequently displaced to the correct positions. This is favorable over triangle strip approaches [18] as quad strips allow a more efficient calculation of the artificial tube normals. Furthermore, the vertex displacement strategy is the same as for lines. The visual-

¹GL_QUAD_STRIP vertex definition for OpenGL

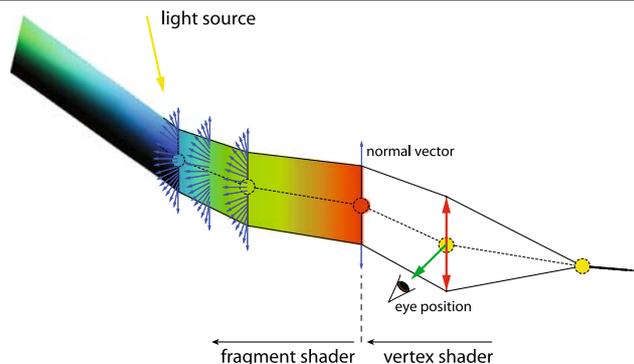


Fig. 6 The development of streamtubes based on a pre-calculated line texture with streamline or pathline supporting points. A vertex shader places the predefined vertices of a quad strip relative to available supporting points and defines an arbitrary color and a normal vector perpendicular to the viewing direction. With the interpolated color and

normal vector values, a fragment shader can perform a Phong-based realistic illumination. The tube evaluation is stopped and therefore interrupted if another previously calculated line segment has already been marked as drawn. This avoids a cluttering of the line segments

ization algorithm can be performed in one vertex shader and one fragment shader, which is illustrated in Fig. 6.

Also a simple but effective technique to represent parts of the field is the insertion of different kinds of planes in a volumetric velocity field. These planes can consist of many vertices organized in a triangulated mesh (*mesh-plane*) or a simple quad bordered by four vertices (*quad-plane*). Both techniques can be used with different benefits as demonstrated in Fig. 9. Many vertices for one plane allow an online deformation according to the flow field but tend to produce self-occlusions themselves and are more expensive to compute than the processing of only four vertices. Quad bordered textured planes are cheaper to compute but they also suppress a considerable amount of information. The usage of velocity directions as normal vectors can be applied to get to a new approach for quad-planes. We use a fragment shader to bring back the missing information. The flow directions within the plane are interpreted as a normal map for the illumination. This technique lets actual flat surfaces appear with a profile. If the user knows that the velocity direction is used as normal vector, she can easily interpret a three-dimensional information into a two-dimensional plane. The crib for example in a scene with a head light will then be: “The more it reflects, the more the flow indicates toward me. The darker it is, the more the flow goes away from me” (see for example the vortex in Fig. 9). The drawback of this method is that it is not fully intuitive and that it requires meta-knowledge. However, our medical partners confirmed that these cutting planed have the benefit that several of them can be combined in the same scene without getting lost in vertex displaced planes.

3.3 GPU accelerated dense visualization algorithms

Based on the former assumptions made for streamlines and pathlines we can also extend the well known line integral

convolution (LIC) algorithm [13] to a four-dimensional interactive visualization. Given a streamline σ , LIC computes the intensity I of a pixel located at $x_0 = \sigma(s_0)$ by

$$I(x_0) = \int_{s_0-L}^{s_0+L} k(s-s_0)T(\sigma(s)) ds, \quad (1)$$

where T denotes the white noise input texture, k denotes the normalized filter kernel of length $2L$ and s is the arc-length [25]. As the length of the filter kernel $2L$ strongly influences the performance of the algorithm, it has to be selected with care. [25] report that a length of 1/10th of the image width returns good results. We came to a similar conclusion, so $2L$ is set to 1/10th of the image width per default. For performance reasons we use a box filter kernel. For streamline computation and to compute the convolution integral, we use an approach similar to the one presented by [25]. To compute the streamline a Runge–Kutta integration of fourth order is applied. For the convolution integral we sample the input texture T at evenly spaced locations x_i along the streamline $\sigma(s)$. The step size between the sample points is stated by h_t and is chosen half the size of a pixel to avoid aliasing. With these assumptions (1) becomes:

$$I(x_0) = \frac{\sum_{i=-n}^n T(x_i)}{2n+1} \quad \text{with } x_i = \sigma(s_0 + ih_t). \quad (2)$$

Using (2) the intensity for each voxel in the 3D volume is computed. The original LIC algorithm by [2] extends naturally to three-dimensional vector fields. The vector fields at different time positions are treated one at a time. However there are a few changes that need to be made. Firstly, not only the vector field, but also the noise-texture have to be three-dimensional. Secondly, the dimension of the volumes has to be taken into account. If it is not isotropic different step sizes h_t for each axis have to be chosen. Further-

more the length of the filter kernel is not 1/10th of the image width but 1/10th of the longest edge.

Our framework allows freely definable views and arbitrary combination of sparse flow visualization algorithms for each view as outlined in screenshot Fig. 7. Therefore, flow, non-flow and other data sets can be compared within and between views. This can be achieved without a loss of interactivity, since flow textures are only assigned to the graphics hardware when they are really needed, and trajectory calculations are partly shared between visualizations and views for the same data sets.

A GPU raycasting algorithm is used to combine the LIC result with the anatomical background. A combination with

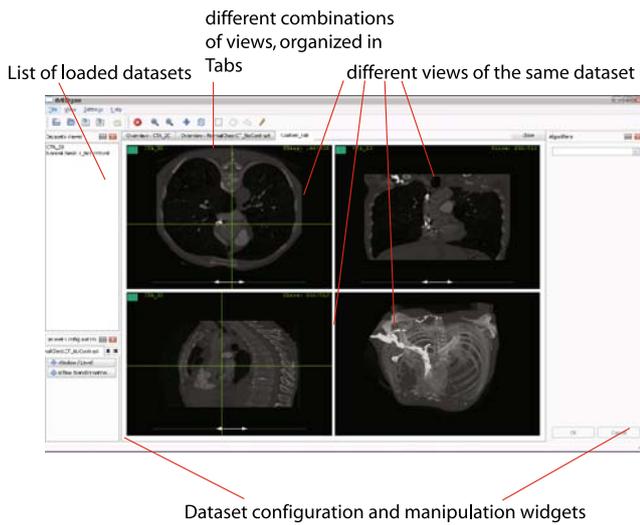


Fig. 7 Our framework allows free definable views for flow or simple anatomic data sets from any modality with arbitrary combination of flow visualizations. Sustaining interactivity is one of the main jobs of the framework. Therefore flow textures are only held in graphics hardware memory if they are really needed. Numerical integration textures or attributes are shared between views and visualization algorithms of the same data set

sparse flow representations in one view is possible, but only of limited use since geometrical representations are not considered during the raycasting approach. Sparse representations therefore lack the correct blending with the LIC and anatomical volume. They appear to be superimposed. Using at least two different views—one for sparse visualizations with plane-based anatomical backgrounds for an assessment of flow details, and one for dense visualizations with raycasting of combined volumes to see the field as a whole is therefore suggested. Figure 8 shows an example of anatomy combined with flow fields as they can be generated with hardware accelerated four-dimensional LIC (GPU-4D-LIC).

For a visual validation of our flow visualization algorithms we built a flow phantom with a flexible tube and a wet-pit pump. Flow visualizations were adapted to these data sets and subsequently applied to human subjects. This phantom and corresponding visualizations are shown in Figs. 9 and 10. Compared to Fig. 11, which shows the flow in the left ventricle at a certain time step, the phantom data

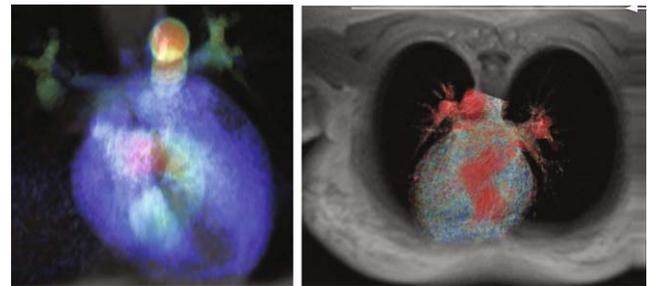


Fig. 8 Both heart chambers can clearly be distinguished in a GPU-4D-LIC view in combination with anatomical background. These two screenshots show the same data set but with different line integration length and different color coding gradients and from opposing sides. Note the dark separation of the chambers and the clearly visible flow in the larger cardiac and pulmonary vessels. Volume rendering is done with hardware accelerated raycasting of the combined volumetric and anatomical data

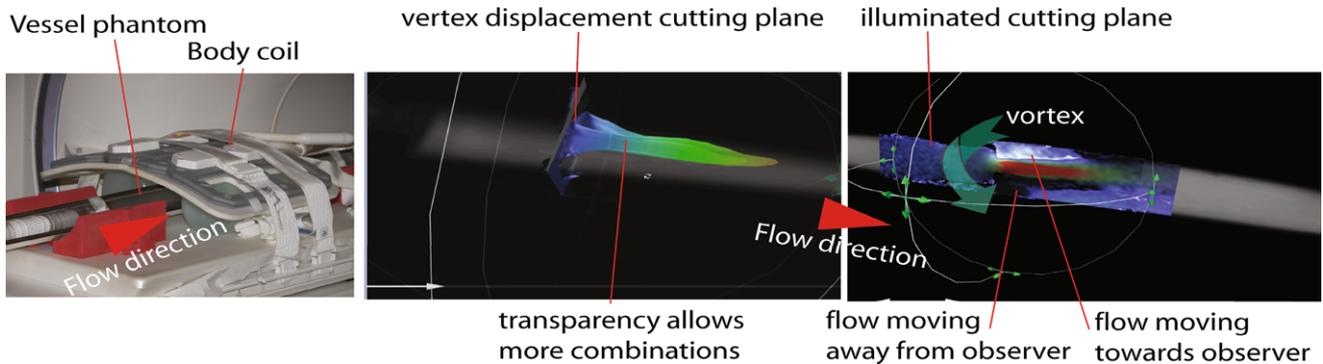


Fig. 9 Comparison of two different cutting plane types applied to a vessel phantom with artificial flow and narrowing. The *most-left* image shows the measurement setup. The *middle* image shows the application of a deformable plane perpendicular to the flow direction. Vertices are

not visible with that visualization or occlude other parts of the plane. Therefore an illuminated but flat plane can be applied in flow direction so that the vortex after the narrowing is clearly observable (*most-right* image)

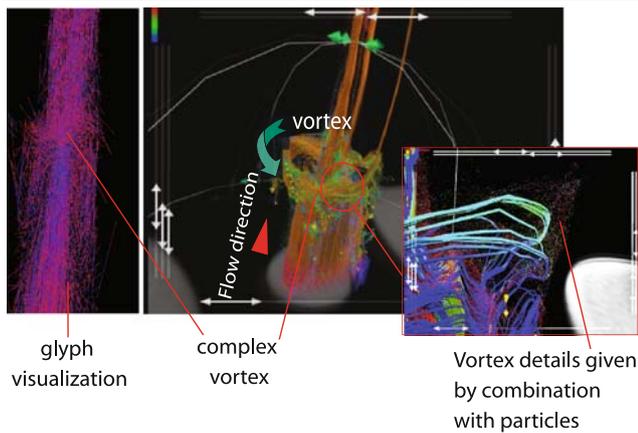


Fig. 10 This part shows an example where a dense glyph visualization inhibits perception of the vortex. The *right* image and its augmentation show how sparse visualizations can provide both, an overview with trajectories and details of the scene with particles and different color maps. The particle trajectories are rendered in that case as direct connection of the supporting points. An interpolation with e.g. B-Splines would provide smoother curves, but would also require more computation time which impairs our aim for absolute interactivity in case of a combination of views

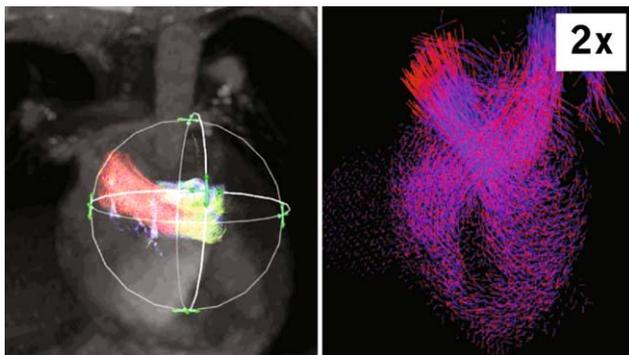


Fig. 11 A screenshot taken from the attached video showing the flow in the left ventricle at a certain time step. The *left* render area shows a particle effect with seed points placed in the left ventricle and morphological MRI data in the background. The *right* render area shows a glyph visualization of the whole velocity field, where every tenth position is shown for a proper overview

sets show stationary parts with a narrowing and the resulting vortex. Due to the complexity of flow in the real vascular system, we use this simple flow constitution for an assessment of different visualization methods and their comprehensible combination.

4 Results

Our framework is currently tested by our medical partners for the usage in clinical routine and clinical research. Up to now they found that each technique is usable for different investigations. Particle and line-based visualizations can be

used for the assessment of small areas within vessels or for the evaluation of vortices which occur during the closure of cardiac valves. Planes, tubes, GPU-4D-LIC and glyph visualizations turn out to be suitable for a good overview over the whole flow velocity field.

Currently our setup is used with a standard personal computer equipped with a GeForce 8800GTX. Interactivity and stability could be obtained during all clinical examples we performed. Videos² showing the stability and interactive capabilities of the framework accompany this paper. Figure 11 outlines a particular section of that video. Thereby a stress scenario for the framework was designed:

- Three loaded MRI flow data sets, each requiring about 200 MB.
- Two CT anatomical data sets (non-flow).
- Eight visualization tabs with each showing at least two different views.
- Six different visualization algorithms combined in one view.

Our framework produced never less than 60 frames per second (fps) during these tests.

Our vessel phantom which shows an artificial but constant flow over time, was able to demonstrate that all tested visualizations yielded a coherent result. All showed a fast flow through the narrowing with the same velocity mapping at different positions using different algorithms and a clockwise vortex around the outlet.

5 Conclusion and outlook

We investigated the visualization capabilities of four-dimensional cardiovascular flow as measurable by magnetic resonance phase-contrast imaging. Beside the improvement and hardware accelerated implementation of known sparse flow visualizations, we developed a framework with freely arrangeable and fully adjustable visualizations.

To evaluate the usefulness of the flow visualization techniques, we built a flow phantom with an artificial narrowing to obtain a clear and concise steady flow over time. We can demonstrate that all developed algorithms yield a coherent *visual* result. Currently we cannot provide a numerical comparison between our phantom measurements and a simulation of the setup. A validation like this would imply a validation of the measurement method itself. This has already been done by [15], so we can assume that we acquire correct flow values for our flow phantom too.

In [22] it was shown that three-dimensional and four dimensional flow patterns can be used for the non-invasive

²Also available at (with XviD Codec): http://www.icg.tu-graz.ac.at/Members/club_kainz//4dFlow/hbbvideo

detection of a cardiovascular disease. More precisely, pulmonary hypertension coincides with the appearance of a vortex of blood flow in the main pulmonary artery. The authors detected mostly visual flow patterns with a simple slice-based arrow plot. The methods as presented in this paper, especially illuminated and time frame marked pathlines or cutting planes with flow dependent reflectance will improve the assessment of these pathological changes in blood flow. Our tools also provide the basics for an automation of this kind of flow pattern analysis.

Furthermore, it is quite likely that these visualization algorithms will allow the perception of additional pathologies correlated to flow patterns. With these techniques and further research on cardiovascular flow, PC-MRI combined with advanced flow visualization techniques will replace risky catheterization procedures with a non-invasive and fast diagnostic method in the long run.

Acknowledgements The research presented in this paper was supported by the European Union in the context of the research project “IMPACT” (FP7-2008-IST-223877) and the Austrian Science Fund FWF under contract Y193.

The presented project was also developed in cooperation with the Graz University Clinic of Radiology providing the data sets, and Siemens Healthcare providing the 4D-Flow preprocessing toolbox for PC-MRI sequences.

Also thanks to our student Felix Nairz who implemented and extended parts of the dense GPU-4D-LIC algorithms for modern GPU hardware and Michael Kalkusch who supported the design of the data flow.

References

- Angel, E.: Interactive Computer Graphics: A Top-Down Approach Using OpenGL, 4th edn. Addison-Wesley, Reading (2005)
- Cabral, B., Leedom, L.C.: Imaging vector fields using line integral convolution. In: SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 263–270. ACM, New York (1993)
- Cebral, J., Castro, M., Appanaboyina, S., Putman, C., Millan, D., Frangi, A.: Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity. *IEEE Trans. Med. Imaging* **24**(4), 457–467 (2005)
- Edelman, R.R., Hesselink, J.R., Zlatkin, M.B. III, J.V.C. (eds.): *Clinical Magnetic Resonance Imaging*, vols. 1–3, 3rd edn. Saunders Elsevier, Philadelphia (2006). URL <http://www.clinicalmri.com/>
- Gonzalez, E., Schoepfoerster, R.: A simulation of three-dimensional systolic flow dynamics in a spherical ventricle: effects of abnormal wall motion. *Ann. Biomed. Eng.* **24**, 48–57 (1996)
- Grabner, M., Laramee, R.S.: Image space advection on graphics hardware. In: Jüttler, B. (ed.) *SCCG '05: Proceedings of the 21st Spring Conference on Computer Graphics*, pp. 77–84. ACM, New York (2005)
- Helgeland, A., Andreassen, O.: Visualization of vector fields using seed lic and volume rendering. *IEEE Trans. Vis. Comput. Graph.* **10**(6), 673–682 (2004)
- Kolb, A., Latta, L., Rezk-Salama, C.: Hardware-based simulation and collision detection for large particle systems. In: *HWWS '04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, pp. 123–131. ACM, New York (2004)
- Kruger, J., Kipfer, P., Kondratieva, P., Westermann, R.: A particle system for interactive visualization of 3d flows. *IEEE Trans. Vis. Comput. Graph.* **11-6**, 744–756 (2005)
- Kvitting, J., Ebbers, T., Wingstroem, L., Engvall, J., Olin, C., Bolger, A.: Flow patterns in the aortic root and the aorta studied with time-resolved, 3-dimensional, phase-contrast magnetic resonance imaging: implications for aortic valve-sparing surgery. *J. Thorac. Cardiovasc. Surg.* **127**, 1602–1607 (2004)
- Laramee, R., Hauser, H., Doleisch, H., Vrolijk, B., Post, F., Weiskopf, D.: The state of the art in flow visualization: Dense and texture-based techniques. *Comput. Graph. Forum* **23**(2), 203–221 (2004)
- Latta, L.: Building a million particle system. In: *Game Developers Conference 2004* (2004). This document is available under the following link: <http://www.2ld.de/gdc2004/>, last visited on January 11th 2009
- de Leeuw, W., van Liere, R.: Comparing lic and spot noise. In: *VIS '98: Proceedings of the Conference on Visualization '98*, pp. 359–365. IEEE Comput. Soc. Press, Los Alamitos (1998)
- de Leeuw, W.C., Wijk, J.J.V.: Enhanced spot noise for vector field visualization. In: *Visualization Conference, IEEE*, p. 233 (1995)
- Ley, S., Unterhinninghofen, R., Ley-Zaporozhan, J., Schenk, J., Kauczor, H., Szabo, G.: Validation of magnetic resonance phase-contrast flow measurements in the main pulmonary artery and aorta using perivascular ultrasound in a large animal model. *Invest. Radiol.* **43**(6), 421–426 (2008)
- Liu, Z., Moorhead, R.J.: Accelerated unsteady flow line integral convolution. *IEEE Trans. Vis. Comput. Graph.* **11**(2), 113–125 (2005)
- Markl, M., Alley, M.T., Chan, F.P., Wedding, K.L., Draney, M.T., Elkins, C.J., Parker, D.W., Wicker, R., Taylor, C.A., Herfkens, R.J., Pelc, N.J.: Time-resolved three-dimensional phase-contrast MRI. *J. Magn. Reson. Imaging* **17**, 499–506 (2003)
- Merhof, D., Sonntag, M., Enders, F., Nimsky, C., Greiner, G.: Hybrid visualization for white matter tracts using triangle strips and point sprites. *IEEE Trans. Vis. Comput. Graph.* **12**(5), 1181–1188 (2006)
- Morris, L., Delassus, P., Callanan, A., Walsh, M., Wallis, F., Grace, P., McGloughlin, T.: 3-d numerical simulation of blood flow through models of the human aorta. *J. Biomech. Eng.* **127**, 767–775 (2005)
- Nerem, R. Jr., Gross, D., Hamlin, R., Geiger, G.: Hot-film anemometer velocity measurements of arterial blood flow in horses. *Circ. Res.* **34**, 193–203 (1974)
- Reiter, G., Reiter, U., Kainz, B., Greiser, A., Bischof, H., Riemüller, R.: Mr vector field measurement and visualization of normal and pathological time-resolved three-dimensional cardiovascular blood flow patterns. *J. Cardiovasc. Magn. Reson.* **9**, 237–238 (2007)
- Reiter, G., Reiter, U., Kovacs, G., Kainz, B., Schmidt, K., Maier, R., Olschewski, H., Riemueller, R.: Magnetic resonance-derived 3-dimensional blood flow patterns in the main pulmonary artery as a marker of pulmonary hypertension and a measure of elevated mean pulmonary arterial pressure. *Circ. Cardiovasc. Imaging* **1**(1), 23–30 (2008)
- Rezk-Salama, C., Hastreiter, P., Christian, T., Ertl, T.: Interactive exploration of volume line integral convolution based on 3D-texture mapping. In: Ebert, D., Gross, M., Hamann, B. (eds.) *IEEE Visualization '99*, pp. 233–240. San Francisco (1999)
- Schoepfoerster, R., Silva, C., Ray, G.: Evaluation of left ventricular function based on simulated systolic flow dynamics computed from regional wall motion. *J. Biomech.* **27**, 125–136 (1994)
- Stalling, D., Hege, H.C.: Fast and resolution independent line integral convolution. In: *SIGGRAPH '95: Proceedings of the 22nd*

- Annual Conference on Computer Graphics and Interactive Techniques, pp. 249–256. ACM, New York (1995)
26. Taylor, T., Yamaguchi, T.: Flow patterns in three-dimensional left ventricular systolic and diastolic flows determined from computational fluid dynamics. *Biorheology* **32**, 61–71 (1995)
 27. Weiskopf, D.: GPU-Based Interactive Visualization Techniques. Springer, Berlin (2007)
 28. Wigstroem, L., Ebberts, T., Fyrenius, A., Karlsson, M., Engvall, J., Wranne, B., Bolger, A.F.: Particle trace visualization of intracardiac flow using time-resolved 3d phase contrast MRI. *Magn. Reson. Med.* **41**, 793–799 (1999)
 29. van Wijk, J.: Spot noise texture synthesis for data visualization. In: SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pp. 309–318. ACM, New York (1991)



Bernhard Kainz is a research engineer and Ph.D. student at the Graz University of Technology. He received his Master degree in Computer Science focused on biomedical engineering, computer graphics and computer vision from Graz University of Technology in October 2007 with highest distinctions, and his B.Sc. in Telematics in July 2005. His research interests include medical imaging modalities and their post-processing, medical pre-operative planning in virtual environments, intra-operative visualization and hardware accelerated simulation of physiological scenes.

He also developed and implemented major parts of the Siemens 4D-Flow Toolbox which was used for data preprocessing in this work.



Ursula Reiter is physicist at the Department of Radiology at the University Hospital Graz. She received her diploma and Ph.D. in physics from the University of Technology in Graz. Her research interests are clinical applications of cardiovascular magnetic resonance imaging and magnetic resonance based analysis of blood flow patterns. Ursula Reiter is the author and co-author of various peer reviewed papers and book contributions in laser physics and cardiac magnetic resonance imaging.



Gert Reiter is employed by Siemens Healthcare and manages scientific magnetic resonance imaging collaborations in Austria. He received his diploma and Ph.D. in physics from the University of Technology in Graz. His current primary research interests are cardiovascular magnetic resonance imaging as well as associated image post processing and physiological modeling. Gert Reiter gave many lectures and teaching courses in mathematical physics, biomedical engineering, medical imaging and image

post processing. He is author and co-author of various peer reviewed papers and book contributions and gave talks on many conferences and meetings.



Dieter Schmalstieg is Full Professor of Virtual Reality and Computer Graphics at Graz University of Technology, Austria, where he directs the “Studierstube” research project on augmented reality. His current research interests are augmented reality, virtual reality, distributed graphics, 3D user interfaces, and ubiquitous computing. He received Dipl.-Ing. (1993), Dr. Techn. (1997) and Habilitation (2001) degrees from Vienna University of Technology. He is author and co-author of over 100 reviewed scientific publications, member of the editorial advisory board of computers & graphics, member of the steering committee of the IEEE International Symposium on Mixed and Augmented Reality, chair of the EUROGRAPHICS working group on Virtual Environments, advisor of the K-Plus Competence Center for Virtual Reality and Visualization in Vienna, deputy director of the doctoral college for confluence of graphics and vision, director of the Christian Doppler Laboratory for Handheld Augmented Reality and member of the Austrian Academy of Science. In 2002, he received the START career award presented by the Austrian Science Fund.

He is author and co-author of over 100 reviewed scientific publications, member of the editorial advisory board of computers & graphics, member of the steering committee of the IEEE International Symposium on Mixed and Augmented Reality, chair of the EUROGRAPHICS working group on Virtual Environments, advisor of the K-Plus Competence Center for Virtual Reality and Visualization in Vienna, deputy director of the doctoral college for confluence of graphics and vision, director of the Christian Doppler Laboratory for Handheld Augmented Reality and member of the Austrian Academy of Science. In 2002, he received the START career award presented by the Austrian Science Fund.