# Adaptive Animations of Vortex Flow Extracted from Cardiac 4D PC-MRI Data

Benjamin Köhler[1], Uta Preim[2], Matthias Grothoff[3], Matthias Gutberlet[3], Bernhard Preim[1]

[1]Dept. of Computer Graphics and Simulation, Magdeburg University, Germany
[2]Dept. of Diagnostic Radiology, Municipal Hospital Magdeburg, Germany
[3]Dept. of Diagnostic and Interventional Radiology, Heart Center Leipzig, Germany
ben.koehler@isg.cs.uni-magdeburg.de

**Abstract.** Four-dimensional phase-contrast magnetic resonance imaging (4D PC-MRI) acquisitions facilitate the assessment of time-resolved, 3D blood flow information. Vortex flow in the aorta or pulmonary artery is of special clinical interest, since it can be an indicator for different pathologies of the cardiovascular system. Qualitative methods commonly employ animated pathlines to depict the time-varying flow. Visual clutter is reduced via vortex flow extraction. Since vortices are often not present during the full cardiac cycle, parts of the animation show an empty vessel or flow that is not of interest. To exploit the given video length more efficiently, we propose Vortex Animations with Adaptive Speed (VAAS), which depend on the time- and view-dependent feature visibility. Collaborating experts considered our technique as useful for presentations, case discussions and documentation purposes. Four diverse datasets are presented in a qualitative evaluation.

## 1 Introduction

4D PC-MRI data describe time-resolved, 3D blood flow information [1] of one heartbeat consisting of systole, when the blood is pumped, and diastole. Advances in recent years increased the clinical applicability by lowering the scan times to a few minutes. However, 4D PC-MRI is usually performed for research purposes due to a lack of standardized evaluation methods and clinical experience with such data.

There are only few physiological occurrences of vortex flow in the aorta and pulmonary artery. Further vortices can indicate different cardiovascular diseases (CVDs), such as malfunctioning heart valves with altered opening characteristics [2]. In addition, near-wall vortex flow is associated with increased shear forces, which might be an important factor in aneurysm development. Therefore, current medical publications investigate systematic occurrences of specific flow patterns in studies with homogeneous patient collectives.

While quantitative data analysis allows to assess the cardiac function and monitor disease progression, qualitative methods support the identification of characteristic flow patterns. Flow animations commonly employ pathlets or particles [3]. The extraction of vortex flow facilitates an easier comprehension of the highly complex 4D data [4]. Though, vortices are mostly not present during the full cardiac cycle. An interactive manipulation of the animation time is possible in flow analysis software, but rendered videos typically use a constant speed. As a consequence, many seconds of the video are potentially wasted by displaying an empty vessel or flow outside the focus. There are other works that established flow simplifications or abstractions [5]. However, a great advantage of flow animations is their intuitiveness. Therefore, we enhance this technique by describing view-dependent *Vortex Animations with Adaptive Speed* (VAAS). Here, time-spending is concentrated on flow characteristics of interest, whereas the rest is time-lapsed.

Our collaborating radiologists and cardiologists appreciate the easier vortex analysis from videos and see the principal application in presentations, case discussions or documentations. We present four diverse datasets in an informal evaluation.

## 2 Material and Methods

This section describes the 4D PC-MRI data, their preprocessing and our work, which can be seen as a histogram equalization for the temporal vortex visibility.

### 2.1 Data Acquisition and Preprocessing

The 4D flow scans were performed in a 3 T Magnetom Verio (Siemens Healthcare, Erlangen, Germany) with a maximum expected velocity ($V_{ENC}$ parameter) of 1.5 m/s. Obtained *phase images* describe the 3D flow direction and strength per voxel. *Magnitude images* represent the absolute flow strength. Spatio-temporal resolutions of about 1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 50 ms are achieved in a grid with 132 rows$\times$192 columns$\times$20 slices$\times$18 temporal positions. *Phase unwrapping* and *velocity offset correction* were performed to reduce artifacts in the phase images (see [5] for further details). A *temporal maximum intensity projection* (TMIP) was performed on the magnitude images to obtain a contrast-enhanced 3D image as approximation of the vessel dynamics. This was used as basis for the subsequent *graph cut*-based segmentation (GridCut library). The binary vessel mask was refined with morphological closing and opening. A triangular vessel surface, which is used for visualization only, was extracted via *Marching Cubes* (VTK library). Pathlines within the vessel mask were calculated with a GPU-implemented *Runge-Kutta-4* scheme and vortex flow was extracted using *line predicates* in combination with the $\lambda_2$ vortex criterion [4].

### 2.2 VAAS: Vortex Animations with Adaptive Speed

A *video* with the target length $S$ seconds and *FPS* frames per second consists of $N = S \cdot FPS$ single images. The cyclic 4D PC-MRI dataset has $T$ temporal


(a) (b) (c) (d)

**Fig. 1.** (a) Animation of extracted vortex flow in a patient with a dilated ascending aorta. (b) View-dependent vessel mask. (c) Feature images (here: binary masks) of extracted vortex flow for each frame of a video with constant speed (red) and (d) corresponding frame-dependent feature visibility function (blue).

positions. For standard videos, the temporal offset between successive frames is constant with $\Delta t(i) = T/(N-1),\ i = 0 \dots N-1$.

The number of pixels in the current viewing direction that show the vessel mesh is required (see Fig. 1(a)). This can be counted on a *binary vessel mask* (see Fig. 1(b)), which is rendered with disabled shading and disabled depth testing into the stencil buffer. Only front face culling should be enabled, as it normally is when intravascular flow is shown.

*Feature images* (see Fig. 1(c)) of the extracted vortex flow are created for each frame of the video with constant speed $\Delta t(i)$. The pathlets are rendered without illumination or halos, but with depth testing against the front face culled vessel. The animation's current time $t$ as well as a user-given parameter $l$, which controls the pathlets' length, are used to calculate the visibility $\alpha_{01}(t, p_t, l)$ for each pathline position with its temporal component $p_t$:

$$\alpha_{01}(t, p_t, l) = \begin{cases} 1 & \text{if } \ t - l < p_t < t + l \\ 0 & \text{else.} \end{cases} \tag{1}$$

When checking $t - l < p_t < t + l$ it is important to consider the cyclic nature of the data. Every value of the discrete, frame-dependent *feature visibility function* $f(i)$ (see Fig. 1(d)) represents the ratio of number of foreground (white) pixels in the feature (vortex) mask of that frame and the vessel mask. Each $f(i)$ is set to $\mathit{max}\ (f(i), h)$ as a correction, where the parameter $h$ indirectly controls the maximum value in the adaptive speed function $\Delta t_{\text{ad}}(i)$. If not restricted, $\Delta t_{\text{ad}}(i)$ can skip very large parts of the cardiac cycle if no vortices are visible. We use $h = 0.05 \cdot \mathit{max}\ (f(i))$ as default. $f(i)$ is resampled to $g(k),\ k = 0 \dots 100 \cdot N - 1$ using a periodic spline (ALGLIB Catmull-Rom). $g(k)$ is normalized via dividing each value by the spline's integral.

To maintain the target video length $S$, the integral of the *adaptive speed function* $\Delta t_{\text{ad}}(i)$ has to equal the dataset's number of temporal positions $T$, as $\int_0^{N-1} \Delta t(i)\ \mathrm{d}i$ does. Each frame of a video with constant speed shows $p = 1/N$ % information of the cardiac cycle. In the video with adaptive speed, we equate

```
p = 1/N;    temp = k_old = i = 0;
for k = 0 to 100·N − 1 do
    temp += g(k);
    if temp ≥ p then
        Δt_ad(i) = T·(k − k_old)/(100·N);
        i += 1;    temp −= p;    k_old = k;
    end if
end for
```

```
p = T/N;    temp = i_old = cnt = 0;
for i = 0 to N − 1 do
    temp += Δt_ad(i);
    if temp ≥ p then
        m = round(temp/p);
        d = m/(1 + i − i_old);
        for n = 0 to m − 1 do
            Δt_diag(cnt) = d;    cnt += 1;
        end for
        temp −= p ∗ m;    i_old = i + 1;
    end if
end for
```

**Fig. 2.** *Algorithm 1* (left): Construction of the adaptive speed function $\Delta t_{\mathrm{ad}}(i)$ from the resampled feature visibility function $g(k)$. *Algorithm 2* (right): Construction of the relative video speed $\Delta t_{\mathrm{rel}}(i)$ from $\Delta t_{\mathrm{ad}}(i)$, which is shown in the diagrams (red).

information with vortex visibility. This is used to transform $g(k)$ to $\Delta t_{\mathrm{ad}}(i)$ via Alg. 1 (Fig. 2). If enough information are gathered ($temp \geq p$) from $g(k)$ after $k - k_{\mathrm{old}}$ steps, $\Delta t_{\mathrm{ad}}(i)$ is set to the corresponding portion of the cardiac cycle. To avoid irritating sudden speed changes, $\Delta t_{\mathrm{ad}}(i)$ is smoothed in two iterations with a 1D binomial filter with kernel size 3 (experimentally determined default). A subsequent correction is applied to ensure that $\int_0^{N-1} \Delta t_{\mathrm{ad}}(i)\, \mathrm{d}i \approx \sum_{i=0}^{N-1} \Delta t_{\mathrm{ad}}(i) = T$. While this integral is smaller than $T$, a value $c$ is added to each $\Delta t_{\mathrm{ad}}(i)$. Afterwards, while the integral is greater than $T$, $c$ is subtracted from each $\Delta t_{\mathrm{ad}}(i)$. We use $c = 0.01/(100 \cdot N)$ as default. This allows to alter the integral by 0.01 in each iteration. Finally, we add $T - \int_0^{N-1} \Delta t_{\mathrm{ad}}(i)\, \mathrm{d}i$ to the last value of $\Delta t_{\mathrm{ad}}(i)$ to achieve the exact target video length $S$. In all our diagrams we show the relative video speed $\Delta t_{\mathrm{rel}}(i)$ (Alg. 2, Fig. 2) instead of $\Delta t_{\mathrm{ad}}(i)$. This is more expressive since it displays the speed level of the video at each time step.

For the final video rendering, we employ a ghosted viewing [6] for the mesh and $\alpha(t, p_t, l)$ (Eq. 2) as pathline opacity function. Order-independent transparency [7] ensures correct alpha blending of the semi-transparent pathlets.

$$\alpha(t, p_t, l) = \begin{cases} \sqrt{1 - |t - p_t|/(2 \cdot l)} & \text{if } \ t - l < p_t < t + l \\ 0 & \text{else.} \end{cases} \tag{2}$$

Our procedure is directly applicable to a region of interest. In case of simultaneously occurring vortices this allows to emphasize one of them. In addition, another flow property, such as the pathlines' velocity, can be mapped to gray scale. The feature images (previously binary masks) then store color values between 0 and 1, which are an implicit weighting. Instead of counting foreground pixels, these color values are accumulated, while the remaining procedure is the same as before. This could also be used to emphasize near-wall vortex flow, which is associated with increased shear forces (wall shear stress). However, disabling line illumination is required to avoid false-positive high feature values.

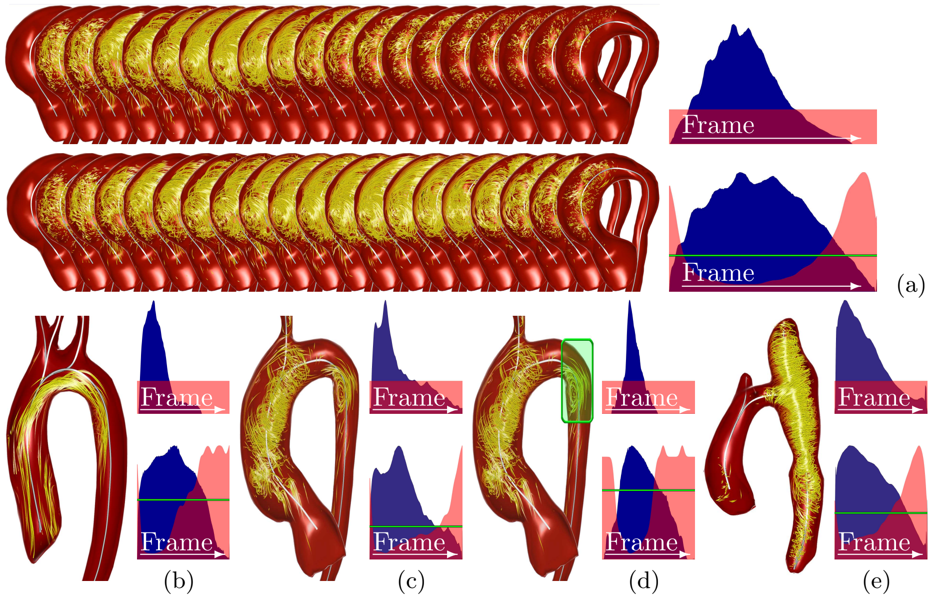


**Fig. 3.** (a) Patient with systolic vortex flow in the ascending aorta. (b) Healthy volunteer with a physiological slight helix in the aortic arch. (c, d) Patient with systolic and diastolic vortex flow in the pathologically dilated ascending aorta and a small vortex in the aortic arch. A region of interest (green) allows to focus the smaller vortex in the aortic arch. (e) Aneurysm patient with a huge vortex that persists during the full cardiac cycle. Diagrams: Corresponding feature (vortex) visibility (blue) with constant (top) and adaptive (bottom) speed (red). The green line in the lower diagram shows the speed level of the video with constant speed as a reference.

## 3 Results

We applied our method to four datasets. The first patient has systolic vortex flow in the dilated ascending aorta (Fig. 3(a)). The adaptive video speed ranged from 19 % to 315 % compared to the constant version. A healthy volunteer with a physiological helix in the aortic arch is shown in Fig. 3(b). In the adaptive video, time lapse is applied to the diastole, where almost none of the extracted vortex flow is visible. Fig. 3(c) shows a patient with pathologic vessel dilation, heavy systolic as well as diastolic vortex flow in it and an additional systolic vortex in the aortic arch. A slight speed increase during diastole is achieved when our method is applied to the whole vessel, since vortex flow is visible during the whole cardiac cycle. Fig. 3(d) shows the same patient with a focus (region of interest) on the smaller vortex in the aortic arch. Due to a shorter overall feature visibility, our method has an increased effect in this case. Fig. 3(e) shows an aneurysm patient with a prominent vortex that is present during the full cardiac cycle. The result is similar to Fig. 3(c).

The employed test system has an Intel Core i7-2600K and a GeForce GTX 680. Our default video setup with 5 s and 50 FPS takes about 50 s to create and analyze the visibility function and another 45 s to render the actual video. The performance directly scales with the desired FPS and the target video length.

## 4 Discussion

We presented Vortex Animations with Adaptive Speed (VAAS) to support the evaluation of cardiac vortex flow patterns. In a figurative sense, the technique is a histogram equalization for vortex visibility. While histogram equalization aims at an optimal use of the available gray values, VAAS aims at an optimal use of given presentation time. The effect, i.e., the use of time lapse and slow motion, is higher the more squeezed the initial feature visibility function is. The results of our method could slightly vary with different line width and pathlet lengths. Also, fast pathlines might be favored implicitly, since they produce longer pathlets and thus occupy more pixels in the feature images. The computational effort could be decreased by not analyzing every video frame to create the visibility function. But, features that are visible for only a short time could be missed this way.

Our method processes only the mesh and pathline geometry and thus is independent from the underlying image data. An application to simulated CFD data or other vessels is conceivable. Our clinical collaborators appreciate the reduced downtimes, where an empty vessel or uninteresting flow was shown. This was considered as helpful during case discussions, presentations and for documentation purposes. It was required that the user should explicitly need to enable our method in the software to avoid confusion. Integrating a speedmeter would further support to recognize the use of time lapse and slow motion.

## References

1. Stankovic Z, Allen BD, Garcia J, et al. 4D flow imaging with MRI. J Cardiovasc Diagn Ther. 2014;4(2):173–92.
2. Calkoen EE, Roest AAW, van der Geest RJ, et al. Cardiovascular function and flow by 4-dimensional magnetic resonance imaging techniques: New applications. J Thorac Imaging. 2014;29(3):185–96.
3. McLoughlin T, Laramee RS, Peikert R, et al. Over two decades of integration-based, geometric flow. Comput Graph Forum. 2010;29(6):1807–29.
4. Köhler B, Gasteiger R, Preim U, et al. Semi-automatic vortex extraction in 4D pc-mri cardiac blood flow data using line predicates. IEEE Trans Vis Comput Graph. 2013;19(12):2773–82.
5. Köhler B, Born S, van Pelt RFP, et al. A survey of cardiac 4D PC-MRI data processing. In: Procs EG Vis Comput Biol Med; 2015. p. 139–48.
6. Gasteiger R, Neugebauer M, Kubisch C, et al. Adapted surface visualization of cerebral aneurysms with embedded blood flow information. In: Procs EG Vis Comput Biol Med; 2010. p. 25–32.
7. Yang JC, Hensley J, Grün H, et al. Real-time concurrent linked list construction on the GPU. In: Procs EG Symp Rendering; 2010. p. 1297–304.