Paper C

Curve-Centric Volume Reformation for Comparative Visualization

Ove Daae Lampe[1,3], Carlos Correa[2],
Kwan-Liu Ma[2], Helwig Hauser[3]

[1]Christian Michelsen Research, Bergen, Norway, www.cmr.no

[2]University California, Davis

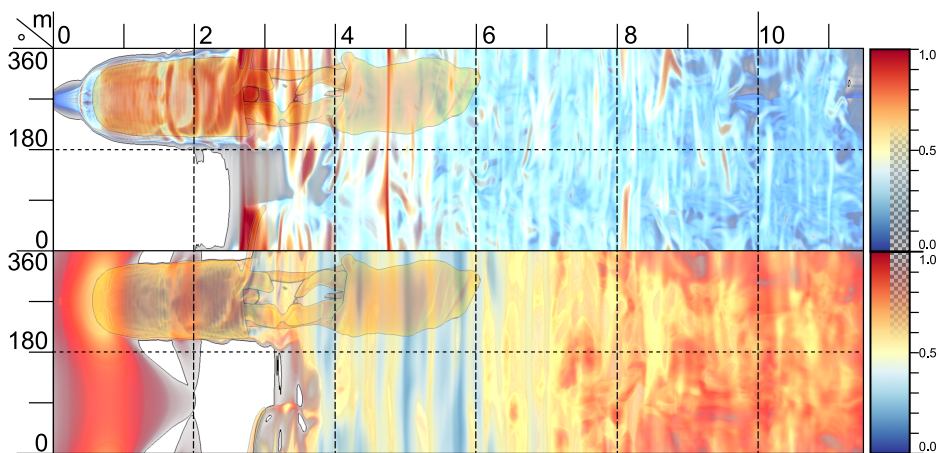[3]Department of Informatics, University of Bergen, Norway, www.ii.UiB.no/vis

Figure 1: The vorticity magnitude on the top and the velocity magnitude on the bottom (volume-rendered in both cases) of a wind simulation around a car as seen from the inside of a streamline and out radially. The horizontal axis is arc-length of the streamline in meter.

Abstract

We present two visualization techniques for curve-centric volume reformation with the aim to create compelling comparative vi-

sualizations. A *curve-centric volume reformation* deforms a volume, with regards to a curve in space, to create a new space in which the curve evaluates to zero in two dimensions and spans its arc-length in the third. The volume surrounding the curve is deformed such that spatial neighborhood to the curve is preserved. The result of the curve-centric reformation produces images where one axis is aligned to arc-length, and thus allows researchers and practitioners to apply their arc-length parameterized data visualizations in parallel for comparison. Furthermore we show that when visualizing dense data, our technique provides an inside out projection, from the curve and out into the volume, which allows for inspection what is around the curve. Finally we demonstrate the usefulness of our techniques in the context of two application cases. We show that existing data visualizations of arc-length parameterized data can be enhanced by using our techniques, in addition to creating a new view and perspective on volumetric data around curves. Additionally we show how volumetric data can be brought into plotting environments that allow precise readouts. In the first case we inspect streamlines in a flow field around a car, and in the second we inspect seismic volumes and well logs from drilling.

1 Introduction

Successful comparative visualizations build upon one or several shared axes as a reference for attributes that should be compared. Tufte called this visual parallels: "Spatial parallelism takes advantage of our notable capacity to compare and reason about multiple images that appear simultaneously within our eyespan" [122]. Many 2D and 3D visualizations often provide a common reference where comparison is readily available. However, there are numerous cases when the quantities being visualized do not conform to a single shared axis or the common reference makes their comparison difficult. For example, production wells, into oil reservoirs, are drilled with complex geometries and turns rather than, previously common, straight vertical wells. In current operations, most preparations, i.e. well planning, etc., are done in 3D environments, whereas the end product, the drill plan, and all drilling data, is produced in 1D, along well length. Current data analysis with regards to measurements from the well, is done in regular graphs along well-length, but this technique is lacking the spatial 3D context, something we address in this paper. Furthermore, the comparison between two wells of disparate shape and length is difficult in the shared 3D space. Instead, it becomes apparent that a more meaningful comparison is obtained when each well is straightened and the visualized quantities are visualized along their arclength. Multiple wells of varying length and shape can now be contrasted in a single shared space. A similar need emerges for the comparison of streamlines in a flow simulation. Since individual streamlines can have an arbitrary shape, quantities such as velocity magnitude and vorticity along the streamline, etc., they cannot be compared directly in 3D space due to inter-occlusion and differences in curvature and length. Instead, we can generate a shared frame of reference that straightens each streamline and lets us visualize a relevant quantity in terms of their arc-length. An example is shown in Figure 1, where we display the vorticity and velocity magnitude of a wind simulation around a car in a 2D plane representing the space around a streamline. The visualization maps the complex 3D shape around a streamline into a 2D view that plots the distribution of vorticity and velocity radially around a given point. With this new type of visualization, one can easily *quantify* changes of a certain variable along a streamline and correlate them to arc-length or radial angle. Furthermore, one can correlate the behavior of a quantity among several streamlines.

To achieve these visualizations, we present a general notion of a *curve-centric reformation*, which maps the space around a 3D curve onto a frame of reference relating to the properties of the curve. We present two forms of such a reformation. In the more general sense of reformation, curve-centric volume reformation is a mapping from the original 3D space to a 3D curve-aligned space, where one axis represents the length of the curve, and the other two are the (adjusted) normal and binormal vectors. Unlike traditional visualizations, where objects and

lines are shown in a given 3D space, curve-centric visualizations depict the given space in the frame of reference defined by the curve.

Another type of curve-centric reformation is *curve-centric radial raycasting*, which defines the mapping to a 2D plane, where one axis (in this paper always the horizontal one) represents the arc-length of the curve and the other axis (here the vertical one) represents the cylindrical angle around the curve. This type of reformation is reminiscent of 3D flattening used for the visualization of virtual colonoscopies [12, 58]. Unlike virtual colon flattening, where shapes and angles are preserved for better diagnostics, our radial raycasting approach preserves distances, essential for a meaningful *quantitative comparison* of variables along the curve. This radial ray-casting therefore produces images of 3D volumes from a novel *inside-looking-out* perspective. As the aim of both these reformations is to accurately portray the neighborhood of the curve along arc-length, their intended use is not directly comparable with existing space deformation techniques, that are designed to create alternative (deformed) views of objects.

In our approach, we use a variation of the well known Frenet frame [43] for creating moving frames and provide an implementation that fits into a general GPU-based visualization system. Furthermore we demonstrate the usefulness of our approach in two scenarios. In one, we use a curve-centric reformation to visualize quantities along the arc-length of log wells for oil exploration. In the second, we visualize the vorticity and velocity magnitude of a wind flow simulation around a car. We show that shape reformation provides hints about the smoothness and curvature of streamlines. These quantities, which can be cumbersome to represent in the 3D view at the same time, can now be provided as a comparative visualization.

We make the following contributions: (1) We present a curve-centric deformation of volume data for the purposes of cross-comparison and easier quantitative analysis. As such, our deformation preserves arc length and orthogonal distances from the center. This is a departure from traditional curve-guided deformations which preserve local shape but not distances. Although useful for generating new views of an object, traditional deformation does not ensure the preservation of quantities essential for meaningful comparison. (2) We present a novel raycasting view that provides unprecedented inside-looking-out views of complex volume data. Current approaches for raycasting of deformed volumes exploit the programmability of contemporary GPUs, but the resulting visualizations remain essentially outside-looking-in views of the data. Our results provide novel views that show quantities of interest along important curves.

2 Related Work

Curved planar reformation. An important issue in visualization is the rendering of complex objects in simpler spaces. Some of these issues stem from the

need to compare and measure areas and distances in a 2D plane rather than an arbitrary 3D shape. One such example is curved planar reformation, which maps a volumetric space around a curve to a plane. This idea has been used for virtual colonoscopy and the visualization of other curved structures. Vilanova et al. use nonlinear raycasting to flatten the internal view of a virtual colon [12]. Similar techniques have been proposed by Wang et al. [131], who unravel the colon using a physics-based deformation of the centerline, and Hong et al. [58] and Haker et al. [51], who use a conformal mapping [58]. These approaches use nonlinear raycasting to compute a 2D inside-looking-out image of the colon. A recent approach by Williams et al. [137] also unfolds the colon using multiplanar reformation. However, their visualization is not a planar mapping, but instead an orthogonal projection of the reformed volume. As such, although less effective as a compact map, the result appears more familiar than the planar warping.

In a similar way to colon flattening, Kanitsar et al. presented (and later improved) curved planar reformation for vascular structures [68, 69]. Unlike the colon, vascular structures exhibit frequent bifurcations, which leads to 2D mappings that branch out along with the vascular structures. The reformation of vascular structures was also explored by He et al. [56], who automate the definition of curves by extracting the medial axis of vessels of interest. Lee and Rasch improved this method by considering topological invariant transformations that lead to better visualizations with little artifacts due to reformation [79]. Curve planar reformation also benefits the visualization of misaligned features. For example, Vrtovec et al. [128] uses curve planar reformation to align the central curve of the spine with the sagittal and coronal planes of the 3D images of the spine. This alignment lets radiologists compare different vertebrae in a single image.

In this paper, we are not bound to a particular mapping of a 3D volume, but rather present a more general notion of reformation, called *curve-centric reformation*, which maps the space around a curve to either a 3D volume or to a 2D plane. This enables us to create novel inside-out visualizations of complex datasets, somewhat similar to planar mappings, from oil well exploration to vehicle design.

Space warping. Curve-centric reformation can be also understood as a type of space warping. Space warping is a general methodology for deforming complex objects by warping the space surrounding them. Because volumetric models often have no explicit geometry, this method is often associated to volume deformation. Some of the first attempts to use space warping to deform objects include Barr's global and local deformations, defined procedurally as geometric transformations [11], and Sederberg's free-form deformation [105], which deforms solid geometric models by warping a tri-cubic lattice enclosing that object. To overcome the need for possible dense control lattices, Sumner et al. [113] uses a graph structure to deform the local space surrounding a number of nodes in an object. Singh explores the use of domain curves or *wires* to deform the space

near them [110]. For a complete survey in space deformation, refer to Gain et al.'s paper [46]. Unlike curve-oriented space deformation, often oriented towards obtaining new poses of a geometric model from a set of user defined curves, our curves are not a means for defining deformation but a centerline along which the user can visualize a certain quantity and the surrounding space.

The idea of space deformation was later continued by True and Hughe [118] for volume warping. More recent volume deformations exploit hardware acceleration to obtain volume deformations in real-time, combining control lattices and volume rendering [133, 99]. As an alternative to proxy-based deformation, one can attain the same results by warping the rays used for volume rendering. Nonlinear ray tracing, for example, as proposed by Gröller [49] enables the rendering of nonlinear spaces such as the visualization of relativistic effects, geometric behavior of dynamical nonlinear systems and visualizing particles in a force field. Continuing this work, Löffelmann et al. generalized this technique on how to define a more abstract camera [81], for use in raycasters. Kurzion and Yagel proposed ray deflectors to accomplish volume deformation, which uses point sources to bend rays as they are traced into the volume [78]. Chen et al. generalize this notion to discontinuous deformations in the form of spatial transfer functions [23]. With the advent of fully programmable GPUs, volume deformation has been embedded directly into the raycasting process, enabling the creation of visualizations that resemble surgical illustrations [25]. Deformed volume raycasting, however, retains the outside-looking-in view common in volume rendering. In our paper, we propose novel inside-looking-out curve-centric raycasting views. For a more extensive description of these and other volume deformation techniques, see Chen et al.'s survey [22].

3 Theory

In this section we first present the basis for our approach, in Section 3.1, which is to create a moving coordinate frame, or a tensor consisting of orthonormal vectors. When this foundation is laid, we continue to present our two different curve-centric reformation techniques, in Section 3.2 and 3.3 before we investigate how we can utilize these in comparative visualizations, in 3.4.

3.1 Moving Coordinate Frames

Creating curve-centric volume reformations relates very closely to the problem of creating a local coordinate frame for every point along a curve. This coordinate frame is a tensor, or in other words, a set of orthonormal vectors for every point on a curve $\mathbf{r}(t)$. There are several techniques generating this frame basis for curves, and what separates them is the different problems they solve. The Frenet frame [43] uses the first and second derivate to create a frame that is

intuitive and gives good geometric insight into the curve itself. This tensor is defined by the curve alone, and leaves no control to the user in modifying the tensor without changing the path of the curve. The Frenet frame is however, not defined for curves where the second derivative is zero, e.g. straight sections, or change of curvature. On positions where the sign of the second derivative changes, the tensor "flips". Klok [75] introduced a solution to these sign changes, by introducing a fixed up-vector, and by restricting the curve to those residing in a plane. If we employ Klok's technique on curves in 3D (even though they were not intended for this), the tensor would collapse when the curves derivative is parallel with the selected up vector, and potentially have a sign change ("flip") afterwards. Another method, that can be used to generate frame tensors, is thin plate splines, as introduced by Duchon [32]. With the analogy of bending sheets of metal, this technique produces tensors along the curve where the radial rotation is minimized. This method produces smooth tensors with no sign changes, but this method leaves, similar to the Frenet frame, no control over the direction of the tensor. In our applications we see that preserving a logical up-vector is producing a better spatial reference. Additionally, since the thin plate splines optimizes the global minimum, minor changes to the curve might radically change the result. Because of this, we introduce a technique that allows the user to specify an up vector, while not exhibiting any sign changes. Our technique is similar to that of Klok [75], but allows curves in 3D and the complete tensor is smoothed to avoid sign changes and high frequency changes in the tangent.

The Frenet frame defines the tensor using local derivatives [43]. Let $L \in \mathbb{R}$ be a positive value and $\mathbf{r}(t)$ a parametric curve that is defined for the interval $t \in [0, L]$. The Frenet frame then defines its axes as

$$\mathbf{T}(t) = \frac{\mathbf{r}'(t)}{||\mathbf{r}'(t)||}, \quad \mathbf{N}(t) = \frac{\mathbf{T}'(t)}{||\mathbf{T}'(t)||}, \quad \mathbf{B}(t) = \mathbf{T}(t) \times \mathbf{N}(t)$$

with $\mathbf{T}(t)$ being the unit tangent, $\mathbf{N}(t)$ the unit normal, and $\mathbf{B}(t)$ the unit binormal. The Frenet frame is quite elegant as it can be explicitly computed for every point on the trajectory, but restricts the selection of curves to only those that are twice continuously differentiable, e.g., no curves with points of inflection. Klok's modified Frenet frame [75] is defined for curves in a plane where the normal $\mathbf{m}$ of the plane defines the binormal $\mathbf{B}_K(t) = \mathbf{m}$, and the tangent of the curve $\mathbf{T}(t)$ and completes the frame by $\mathbf{N}_K(t) = \mathbf{B}_K(t) \times \mathbf{T}(t)$. The Frenet frame, Klok's modified Frenet frame, and thin plate splines, place the tangent component of the vector, $\mathbf{T}(t)$, equal to that of the curve's tangent. Figure 2 shows a 2D version of a straightened curve in two versions, the upper using the tangent as a basis for creating the normal, and the lower using a constant normal. In this example this makes a drastic difference on both the readability and the mathematical complexity of the result. Following this we introduce a smoothing
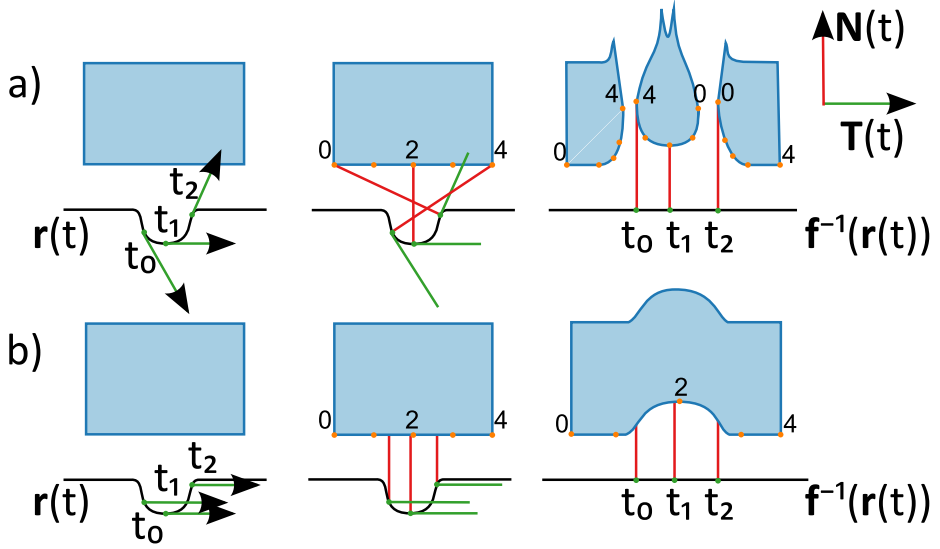
Figure 2: The blue box represents a volume, and the black line the curve to straighten. Two rows showing a) curve-linear deform with tangential $\mathbf{T}(t)$ and b) the same curve-linear deform, but here with constant tangents. Columns left to right show, tangents, normals, and lastly the deformed box.

kernel K_h, with width h. This smoothing kernel is then applied to the original tangent by the following convolution:

$$\hat{\mathbf{T}}(t) = (\mathbf{T} * K_h)(t) = \int_0^L \mathbf{T}(\tau)K_h(t - \tau)\,d\tau$$

When applying this convolution to create the smoothed vector $\hat{\mathbf{T}}(t)$ we can experience, with some selections of K_h, a degeneration, where this vector, in a worst case scenario, can evaluate to a zero vector. To avoid this we apply the above convolution/weighted sum after converting the vectors to quaternions.

Another property we would like to translate to the deformed volume is a sense of the up direction. Many datasets have logically defined up/down directions, and in these cases we consider a "desired up vector" $\mathbf{u}$. In all other cases, a consistent vector is chosen instead, as we would like comparable results over several curves. With this vector $\mathbf{u}$, we can now define our *new* normal $\mathbf{N}_m(t)$ and a binormal $\mathbf{B}_m(t)$ according to the rule,

$$\mathbf{B}_m(t) = \frac{\mathbf{u} \times \hat{\mathbf{T}}(t)}{||\mathbf{u} \times \hat{\mathbf{T}}(t)||} \quad \text{and} \quad \mathbf{N}_m(t) = \hat{\mathbf{T}}(t) \times \mathbf{B}_m(t)$$

This rule is very simple, and it is very similar to the one proposed by Klok [75], but unlike his technique this one is not constrained to curves residing in a plane.
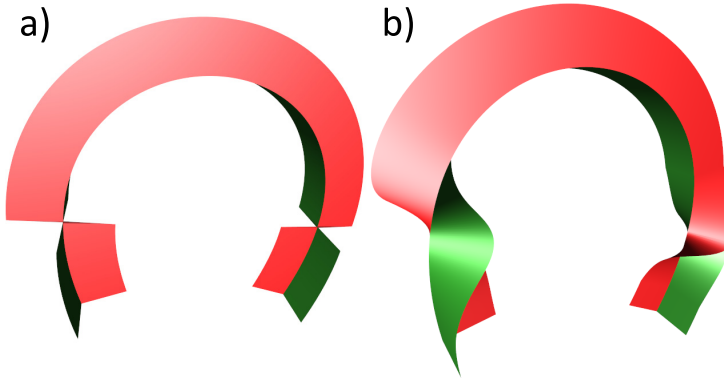
Figure 3: Surfaces following the normal $\mathbf{N}(t)$ as red, and the binormal $\mathbf{B}(t)$ as green. Figure a) shows the modified Frenet frame, experiencing a sign change that the smoothed version, b), handles gracefully.

However, it has two major issues that need to be addressed. The first issue is the obvious case when $\hat{\mathbf{T}} \parallel \mathbf{u}$, and the cross product is 0, accordingly. The second issue is to avoid a sudden sign change of the normal and binormal that can follow such a parallel segment. We solve both these issues by applying a similar smoothing kernel as with $\hat{\mathbf{T}}(t)$. Even more important here than with the tangent, because of previously mentioned sign changes, spherical interpolation must be applied for smoothing, achieved by kernel averaging on the tensors quaternions. From this definition of $\hat{\mathbf{B}}(t)$ we define our normal vector as $\hat{\mathbf{N}}(t) = \hat{\mathbf{T}}(t) \times \hat{\mathbf{B}}(t)$. Figure 3 shows how this smoothing applies to a curve that experiences two sign changes with the modified Frenet frame.

We have now defined a function creating a tensor of orthonormal vectors at each point $\mathbf{r}(t)$ along the curve. In the next section we present two techniques that rely on such a moving frame. To ease the notation and to present that the next techniques are independent of a specific form of frame construction, we will hereby refer to the tangent as $\mathbf{t}(t)$, the normal as $\mathbf{n}(t)$ and the binormal $\mathbf{b}(t)$. However, it is fair to assume that for all practical applications, these vectors are the same as the smoothed versions introduced in this section.

3.2 Curve-Centric Reformation

In curve-centric reformation we aim to create a new volume, parameterized by a curve's arc-length and two terms indicating neighborhood in a distance preserving manner. Unlike several other deformations, the goal of curve-centric reformation is neither to preserve shape, nor angles of external objects, but to correctly portray distances to features and to use these features to provide a spatial frame of reference for the curves trajectory. As shown in the previous section, we make
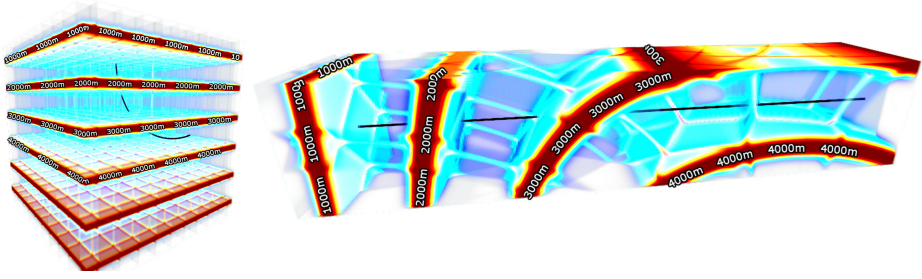
Figure 4: Left, a test volume with a curve in it, and right the result of Curve-Centric reformation. This curve is, after the reformation, the straight line shown in the middle of the volume to the right.

sure that no smoothing is applied to the curve's position, as we would like to strictly enforce these positions as the center of the resulting volume.

We define this reformed volume Y by means of a mapping to the original volume X, or a function $\mathbf{f} : Y \to X$. Given a curve $\mathbf{r}(t)$ and a defined frame set for this curve, we utilize only the normal $\mathbf{n}(t)$ and the binormal $\mathbf{b}(t)$ in the construction of $\mathbf{f}$. We thus define the inverse transformation, from our reformed volume and back to the original as:

$$\mathbf{f}(x, y, z) = \mathbf{r}(z) + x\,\mathbf{n}(z) + y\,\mathbf{b}(z), \quad \text{for } z \in [0, L] \tag{1}$$

which satisfies our initial definition of $\mathbf{f}(0, 0, t) = \mathbf{r}(t)$. It can easily be proven that for any curve other than a perfect line, this mapping is not one-to-one, and will usually contain singularities. An important property of this transformation, besides that it preserves the distance of arc-length, is that it preserves distances orthogonally out from the center, i.e., given two points $\mathbf{p}_1, \mathbf{p}_2$ in a given plane orthogonal to Y's z plane:

$$||\mathbf{p}_2 - \mathbf{p}_1|| = ||\mathbf{f}(\mathbf{p}_2) - \mathbf{f}(\mathbf{p}_1)|| \tag{2}$$

the right hand term is, by decomposing $\mathbf{p}_1$ to $[x_1, y_1, z_1]$, and similarly with $\mathbf{p}_2$, equal to:

$$|| \left(\mathbf{r}(z_2) + x_2\mathbf{n}(z_2) + y_2\mathbf{b}(z_2)\right) - \left(\mathbf{r}(z_1) + x_1\mathbf{n}(z_1) + y_1\mathbf{b}(z_1)\right) ||$$

which we reduce to, since $z_1 = z_2$:

$$||\mathbf{f}(\mathbf{p}_2) - \mathbf{f}(\mathbf{p}_1)|| = ||(x_2 + y_2) - (x_1 + y_1)||$$

since, our initial condition stated, the two points ($p\mathbf{p}_1$ and $\mathbf{p}_2$) share their z component, the left hand side of Eq. 2 can also trivially be reduced to the same term.

94

From Eq. 1 it follows that the Z plane denoted by z in Y equals a plane in X defined by the point $\mathbf{r}(z)$ and the tangent vector $\mathbf{t}(z)$. We utilize this planar coherence to create a fast implementation of curve-centric reformation by using *render to 3D texture*, rendering slice by slice fetching samples from X. Rendering these slices z plane after plane, $\mathbf{n}(z)$ and $\mathbf{b}(z)$ is constant per plane, and the evaluation of $\mathbf{f}$ is done after selecting a width w around the curve to straighten, by linear interpolation between four points, $\mathbf{f}(w, w, z)$, $\mathbf{f}(-w, w, z)$, $\mathbf{f}(-w, -w, z)$, and $\mathbf{f}(w, -w, z)$ with sufficiently large w. The result after such a reformation, is a regular 3D texture which can be visualized using, e.g., a regular volume ray-caster as shown in Figure 4.

3.3 Radial Ray-Casting

As presented in the previous section, curve-centric reformation creates a new regular volume, which has to be visualized using a particular method; we now introduce a direct projection of the original volume called radial curve-centric ray casting. Given a function that creates a unit normal $\mathbf{n}(t)$ and a unit binormal $\mathbf{b}(t)$ from a curve defined by $\mathbf{r}(t)$, traversing the arc-length t we cast rays starting at $\mathbf{r}(t)$ in the direction

$$\sin(\phi)\,\mathbf{n}(t) + \cos(\phi)\,\mathbf{b}(t), \phi \in [0, 2\pi]$$

In its simplest form, by using a straight line, this radial ray-casting technique is reduced to a cylindrical projection, which form a projection that lies in between perspective, having a viewpoint, and an orthogonal projection, having a plane of view, with a line as its starting point. Using this technique one creates 2D projections that have no perspective distortion in the horizontal direction, i.e., the direction along the line, but with perspective distortion in the vertical direction, i.e. the angular rotation around the curve. However in addition to the perspective distortion, for curves, other than those of a perfect line, the curvature of this curve, or rather the torsion of its moving frame tensor, will further distort objects and features as seen from the curve.

A common strategy for ray-casting is to first render the texture coordinates of the exit points for rays, and then to store them. Next the ray casting is initiated by rendering the starting location of these rays, and iterating the ray-casted volume using the line between start and exit. This technique is depcited in Fig. 5, where the left part shows a box used as proxy-geometry for rendering the entry and exit buffer. The strength of this technique is its simplicity and that it is independent of perspective and proxy geometry, and as we will show, also compatible with our radial ray-casting. Generating the exit points is usually done by rendering the proxy geometry, culling front faces, but we will use a
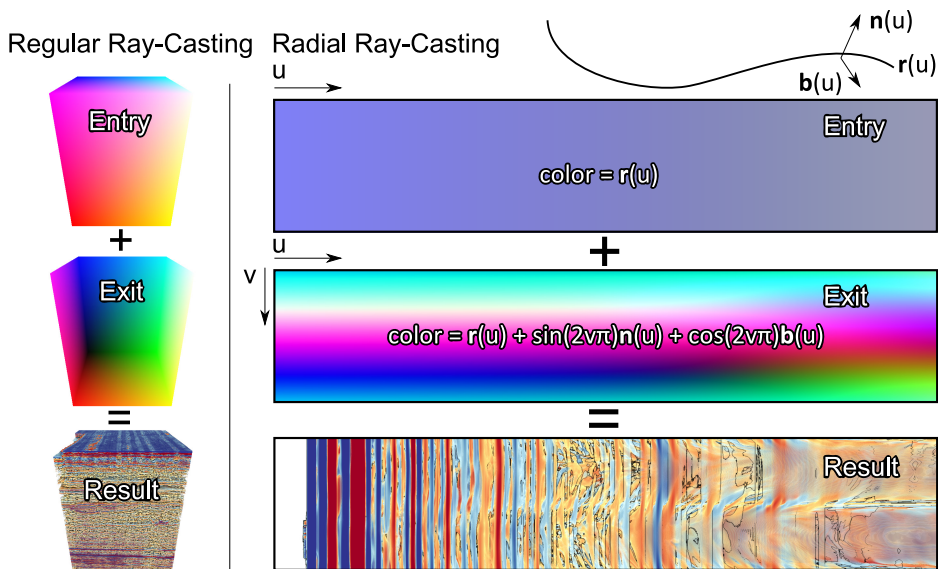
Figure 5: Regular ray-casting uses proxy geometry, generating the entry and exit positions for ray traversal. Our radial ray-casting technique mimics this behavior by setting the entry buffer to the curve's position, and the exit buffer as the curve's position plus an angular rotation around this curve.

more direct method. The exit points in radial ray-casting are defined as, over normalized screen-space u, v, using a far "plane", or tube, with radius far:

$$\mathbf{r}(uL) + \ far \cdot (\sin(2v\pi) \, \mathbf{n}(uL) + \ \cos(2v\pi) \, \mathbf{b}(uL))$$

For efficiency concerns, this exit point should also be clipped, at the intersection point to the texture cube, to stay within existing volume coordinates. After this exit buffer is established, we can start our radial ray-casting, using the start position $\mathbf{r}(uL)$. This start/entry and exit buffer is shown, on the right of Fig. 5. Given the exit position buffer and the starting position, this radial ray-casting technique is theoretically compatible with shader code written for any other ray-caster employing the same strategy, and will thus render at real-time speeds, and be able to perform advanced dynamic transfer functions. When applying proven shader techniques to this type of rays, one of course needs to keep in mind the aliasing issue, produced by the fact that the projection will sample the surrounding space with an uneven number of rays, due to the curvature, and the combination of almost parallel rays in the direction of the curve and an extreme perspective, 360 degrees (like in fisheye projections), in the rays orthogonal to the curve. Figure 1 (the teaser image) shows two examples of radial ray-casting, where a regular color map transfer function has been used, in addition to edge
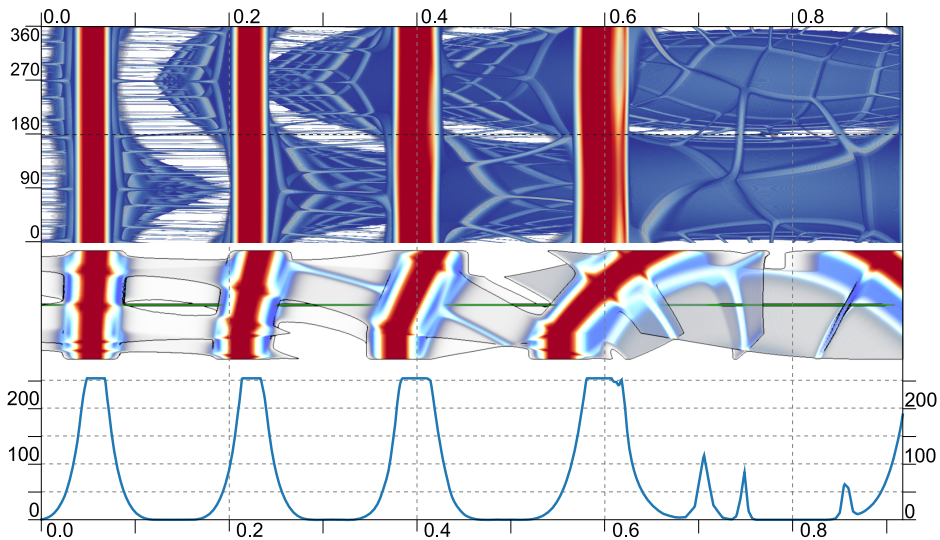
Figure 6: The curve from Figure 4 shown as a curve-centric radial projection, top, as a reformed volume visualized, middle, and a curve plot, showing the sampled intensities from the volume along the curve. This comparative visualization allows accurate comparison of intensity values to their spatial origin.

enhancement. Another example is shown in Figure 6, where the top image shows a radial ray-casting, and because of properties in our generation of the moving frame, we can read out the direction around the curve, as 0 degrees corresponds to the left of the curve, looking in the direction of the trajectory (left to right), 90 degrees up, 180 right and 270 is below the curve.

Unless the curve is dynamic, this rendering produces images with a fixed "view", but one can still implement an interactive exploration of the volume around the curve, by constraining to either a subsegment of L or a subset of the angular radii. An interaction scheme would then allow for zooming, a reduction of segments in both dimensions, and panning, which would translate this segment.

3.4 A Common Axis for Comparative Visualization

Two objects sharing one or more axes have a basis for comparison. A shared axis is simply one that has the same unit, e.g., two physical objects share size, and can thus be compared in terms of size. Moreover, the display of those shared axes should be shown in scale to each other, optimally sharing the exact same scale, to avoid producing deceptive visualizations. In this paper, we present two techniques that create visualizations, by deforming or projecting space, that are aligned with the arc-length of a curve in space. The strong rationale for pursuing

this alignment is to be able to create visualizations that combine well with existing techniques, like graphs, 2D plots, or even images. These are techniques that are not always well imported into a 3D environment, e.g., Figure C.7(a) shows 1D graphs for multiple wells drawn in 3D space, enabling coherence between values and their spatial position, but is not suited for accurate readouts. By reversing this strategy we instead investigate how well we can display information from volumetric models with a correct alignment to the 1D space of these 1D and 2D techniques. One way to look at this reformation is that one renders the physical space in the logging space, instead of rendering the logs in the physical space. A strong argument for this reformation, in Figure C.7(a)'s case, is that the function shown on a single well here is actually only one of several production related parameters that the production team is interested in analyzing. By straightening a single well, all of the different production values can be shown in its own graph, without the inherent occlusion problems that would exist in a 3D visualization. Another argument for exactly this reformation of space, is to provide accuracy in display, e.g., in a plotting environment one has a direct readout of values. Figure 6 shows such a plotting environment, where the two separate displays share the horizontal axis, and one could theoretically use a ruler to match peaks in the bottom curve to features in the two above. The curve-centric radial projection makes sure to preserve a required orthogonality in the direction of the 1D axis of the plot. This orthogonality is also enforced in the visualization of the deformed volume by simply rendering it with an orthogonal projection.

The curve-centric visualization techniques are only fully utilized when actually shown in the same space as important arc-length parameterized metrics. It is not within the scope of this paper to create other novel visualization of these metrics, but to show how one can create comparative visualizations by combining deformed and curve-centric projected, with existing and well established techniques. In the next section we show several examples of arc-length parameterized data, but these are just some of the many that actually exists. As good examples for these data sources, we can imagine most of the sensing devices that are attached to moving objects. These sensing tools are logging data by time, but indirectly they are also portraying information about their surroundings as a function of where they are. Other data sources are, e.g., 1D simulations or trajectory simulations.

4 Application Cases

In this section we show how we successfully applied our techniques to cases of two different industries. In Section 4.1 we investigate two cases with application to the petroleum industry; the first case where we show how curve-centric reformations can help show the multiple data sources gathered when drilling exploration wells; and in the second, we show its application in a real-time drilling scenario, looking

closer into well-bore uncertainty at a certain strategic depth, motivated by an actual incident. In Section 4.2 we similarly investigate two cases with application to the car industry. In both of these two we investigate parameters around streamlines, but in the first one we investigate multiple parameters of a single streamline, whereas in the second we compare several different streamlines on a single parameter.

4.1 Well-Centric Visualizations for the Petroleum Industry

The petroleum industry uses drilling in part for exploration and in part for creating or expanding production in oil and gas reservoirs. Where vertical wells are almost the norm for exploration wells, an increasing number of horizontal wells and even those with more complex geometry, are used to extend the lifetime of aging reservoirs. While drilling these wells, operators apply several sensing tools to learn about the surrounding formation, to get situational awareness, to predict potential problems, and to maximize the drilling parameters, e.g., rotation speed, how much pressure should be applied to the drilling bit, and more. These sensing tools can measure down-hole pressure, electric conductivity of the formation wall and a number of other physical parameters. These provide output with a timestamp, or streaming data in time, but for many of these physical parameters, different representations make more sense. E.g., a sensing device measuring the physical properties of the rock outside the drill-string will provide a measurement as a function of measured depth (arc-length, not true depth) and to a lesser extent rotational position, rather than time. Similarly there are many other parameters that are best shown as a function of depth rather than time graphs, e.g., rate of penetration, hook-load, electric logs, and more. When these logs, all having the common axis of arc-length, are shown to explore their relation to seismic volume data, existing techniques usually put these logs into the 3D seismic space, as shown in Figure C.7(a). We argue that this technique highlights the seismic volume as the important feature, showing the logs in a contextual manner. We propose to invert this display, and to show the seismic in the space of the logs instead. This is done by a reformation of the volume around the wells, and thus the seismic volume share arc-length as the axis with logs. An example of this is shown in Figure 8 where different measures along the drill-string are compared for coherence, as is a common operation in exploration drilling. This figure shows a volume visualization of the deformed volume on top, containing the well, an orange line, in the middle. The next is a graph of seismic reflectance sampled along the wellpath. As expected we see a strong correlation between this and the one above and below, which is the radial ray-casted volume as seen from the well. The final image, at the bottom, is a physical measured value, called an UBI image, or Ultrasonic Borehole Image. This imaging device produces 360 degree images of the formation around the well, very much like our radial raycaster, and we would thus expect to see a very tight correlation be-
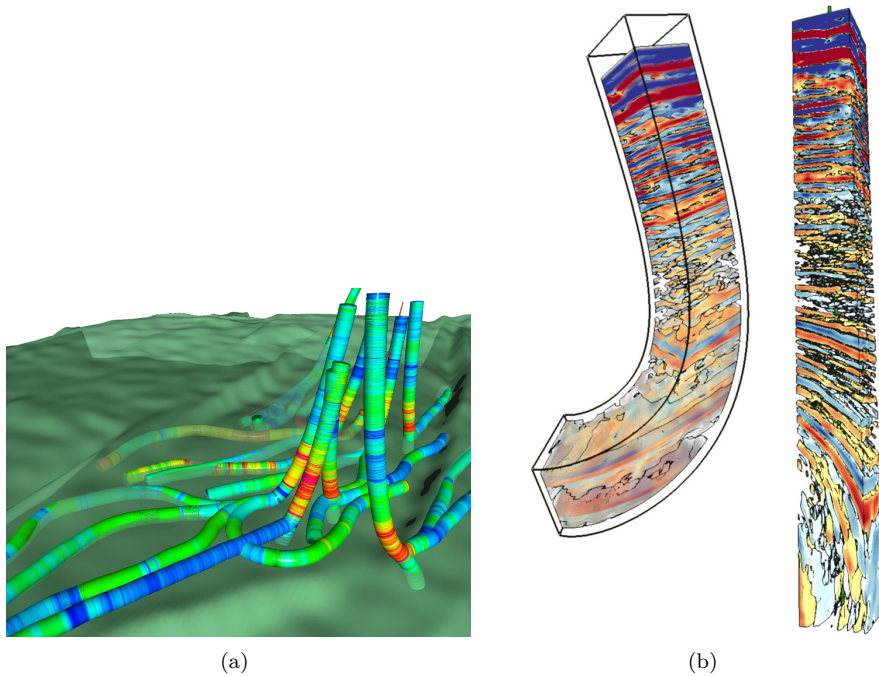
(a)   (b)

Figure 7: (a): Multiple production wells shown as tubes, where color indicate a single physical measured variable along the well length. All these wells reside in the reservoir, some injecting fluids while others receiving fluids. Image used with permission by StatoilHydro ASA. (b): The well we investigate in this case. This figure shows the seismic volume in the close vicinity to the well, before and after reformation.

tween these. Initial seismic volumes are often quite uncertain, and to minimize these uncertainties they are compared to results from exploration wells. We see the techniques of curve-centric reformation as a natural addition to the already existing techniques for studying these results.

Real-Time Drilling Data is the second case in which we introduce a novel view on drilling. The contractors that drill wells have to follow a strict plan of where the drill string should be on the way to the target. In the preparation, to create a drill plan, the close proximity to the well is mapped out and simplified into a 1D plan of different properties that the wellbore passes through. These properties are, e.g., pressure gradients, lithology (rock type), and where stops, logs, and casings should be positioned, all of which share the axis of arc-length. A case using some of these drilling properties is shown in figure 9, in which we inspect the properties of the well as shown in Figure C.7(b) before and after reformation. In this case we show an important part of a drilling operation, in which we are drilling the final stretch before we enter the reservoir, denoted by the
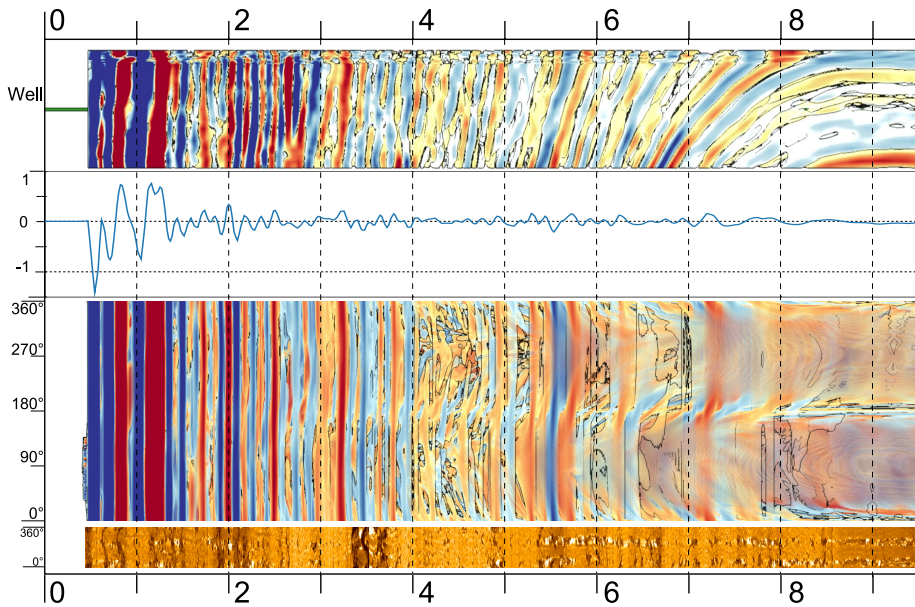
Figure 8: The above images show different measures along a well-path. From top, an orthogonal ray-casted view of a curve-centric deformed seismic, second graph shows the volume intensity (seismic reflectance), third shows an angular ray-casted view from the wellpath and out into the surrounding seismic, and lastly the final image shows an ultrasonic borehole image, an angular view into the formation.

blue limestone. Because of very different properties of the shale (green) and the limestone it is very important to stop drilling as close as possible to this horizon, and insert a casing to protect the wellbore and formation from changing pressures. In addition to providing spatial awareness, this figure highlights an important detail our petroleum industry partners has expressed interest in, namely the wellbore uncertainty. The red ellipse, shown in the same figure (9), represents the area, or rather the volumetric ellipsoid, in which the drill-bit is positioned with 95.4% accuracy. Notice that the 1D lithology column shows the limestone to start at the wellbore's intersection with a major horizon at 5.45 which is diagonal. It is fair to assume that everything below this diagonal horizon is included in the reservoir. This shows that the 1D lithology is only accurate if the wellbore traverses where it was planned to. The 1D lithology column fails to incorporate the uncertainty of wellbore positioning, eg., the wellbore could be positioned deeper, and the reservoir would encountered earlier. In fact, the ellipse shows that given a 95.4% probability safety margin, one *cannot* guarantee that we have not already entered the reservoir. Projecting this ellipsoid into the
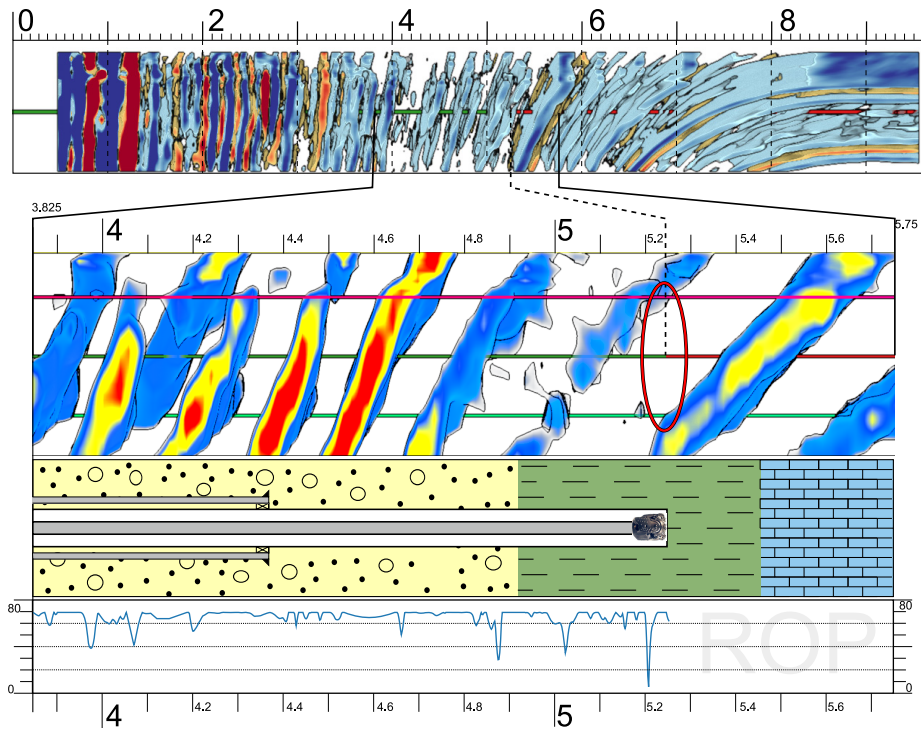
Figure 9: Deformed seismic can provide spatial reference for real time drilling data as well as showing uncertainty in the 1D lithology column. Image on top is the full length deformed wellbore, below a zoom-in of the section currently drilled, which also contains the 1D lithology with the current drill bit position, and a real time graph showing the rate of penetration (ROP) for the section. The red ellipse in the center shows positional uncertainty.

1D lithology column would not reveal this. As mentioned earlier, this example is motivated by an actual incident, where wellbore uncertainty led to the wrong assumption that one had a good clearing before entering the horizon. Secondary measures (gas show) did in fact even show signs that the reservoir was entered, but was ignored due to the believed clearing, and drilling was resumed. We contest that if our display would have been used in this case, then the secondary sign would not have been ignored, since the probability is already shown to be below the threshold for entering the horizon.

4.2 Streamline-Centric Visualization

In this section we investigate a dataset containing velocity and pressure from a single time step in a wind simulation for vehicle design. Additionally we created

a distance field from a geometry file. From the velocity field, we extracted two scalar fields, velocity magnitude and vorticity magnitude. An overview of these datasets are shown in Figure 10. From the original velocity field we further extracted several streamlines, from which we selected a few streamlines that shared an interesting property in that they traverse in close proximity to a side mirror, an overview of these streamlines are shown in Figure 11. Inspecting one of these streamlines is enabled in our shared axis view, where multiple views on the streamline are provided for comparison. This comparative visualization display, as shown in Figure 12, is well suited for understanding how the streamline is affected by the different fields, in addition to looking at the correlations between the different fields. The deformed car on the top of Figure 12 acts as a spatial reference, additionally to reveal information on the curvature, the higher the curvature the bigger the deformation on the car. In this view (still Figure 12) we can see that there is a very good correlation between velocity magnitude and pressure in front of the car. At 1.5 meters into the streamlines arc-length, the streamline passes through a positive pressure, which is aligned with a slight drop in velocity. More interesting is the low pressure the streamline passes through at 2 meters, which interestingly enough does not seem to affect the velocity. Right behind the car, at 5.5 meters, we can see a drop in velocity (the blue vertical feature), that does not seem to correspond with any of the other views. One possible explanation might be the vorticity and low pressure feature right in front of it. Studying correlations in this manner does provide a new perspective into the study of flow fields that our industry partners found intriguing.

Our application partner, a team of engineers who use computational fluid dynamics simulations for all aspects of automobile design, was intrigued by the alternative views that we are able to create with deformation and the curve-centric radial projection. The concept of applying deformation on the car body to reveal a secondary effect outside of the car is very enticing to them. This deformation would then be a tangible communication between the bodywork designers and them for shape optimization. In particular, they pointed out that by choosing the appropriate flow quantities other than velocity, such as vorticity vector or helicity density, the deformed surface could actually suggest the location and extent of the shape change needed to achieve optimal performance. Additionally, the curve-centric radial projection of the flow structure with respect to a straighten field line may display information hard to reveal with conventional flow visualization techniques, such as drag force acting on the car.

5 Summary and Conclusion

In this paper we have presented a general solution on how to create new curve-centric parameterizations of volumetric space. Additionally we have presented how to use this new parameterization in creating two visualizations that are
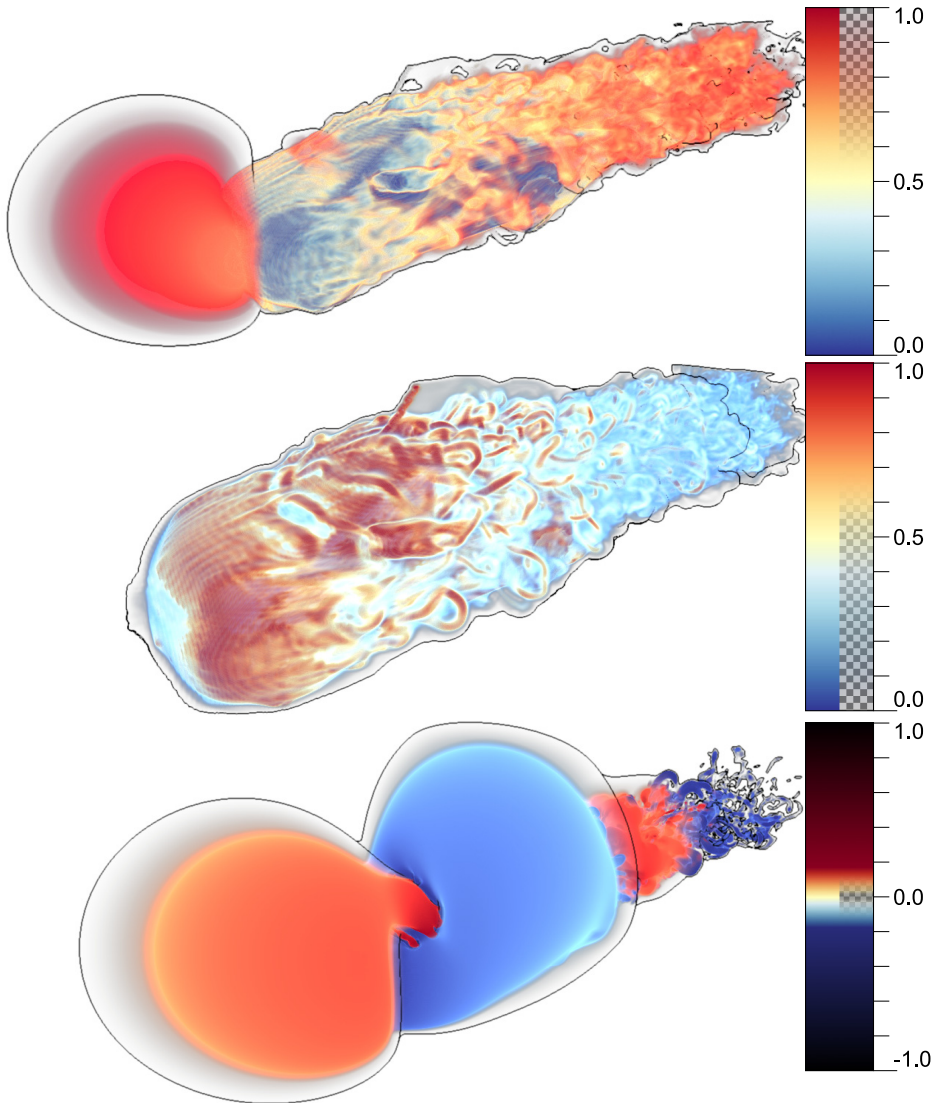
Figure 10: Top, velocity magnitude, middle, vorticity magnitude, and below pressure, all from a vehicle design simulation.

aligned with the arc-length of the curve. We have shown that we can use this alignment to create comparative visualizations, where 3D spatial positions are shown directly overlapping with 1D, or 2D arc-length parameterized functions.

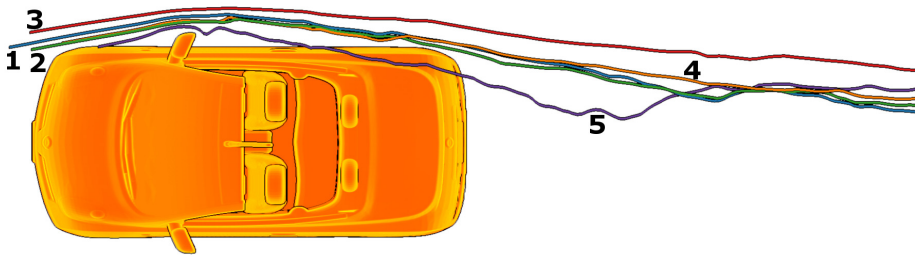We have successfully created a prototype using a combination of C++ and

Figure 11: Showing five selected streamlines with a close proximity to the side mirror, that provides interesting features to study.

Python, which implements the creation of a moving frame given a curve, an optional up vector and a smoothing factor. Using this moving frame and supplying a volume, 3D texture or array, and a width, the prototype creates a deformed volume either as a 3D texture or as an array. This prototype supplies methods for creating images rendered either offline or real-time of both the deformed cube, as an outside in view, and the radial 2D projection, giving inside out views. Using this prototype we have created two case specific applications, one to investigate streamlines, and one to investigate well data from the petroleum industry. To avoid reinventing existing visualization techniques we have integrated our techniques with matplotlib: A 2D Graphics Environment as presented by Hunter [59], which enables the use of existing 1D and 2D visualization techniques, along the aligned axis of our result. Our test system has an Intel Core2 Quad CPU and a GeForce 8800 GTX. Creation of a deformed volume with dimensions [128,128,512], from an original volume [600,300,750], takes 19 milliseconds. Rendering a radial ray-casted image with dimensions [2048,512] of the original volume takes 51 milliseconds.

Limitations and Future Work: The presented algorithm requires a user specified up vector and smoothing factor, and for the volume deformation, also a user specified width. While this user input provides flexibility, it does represent a limitation for complete automation. There is a correlation between the frequency of changes of the curve, the width of the box surrounding the curve, and the smoothing factor. This correlation is not explored in this paper, but could potentially give some interesting automation of these parameters. The radial ray-casting technique does not duplicate any of the original data, but the volume deformation does. When straightening curves with high frequency changes, which will then have a large arc-length within a small section of the original volume, this duplication of voxels becomes very apparent. Another limitation, is when sections of the curve, larger than the smoothing kernel, are parallel with the up-vector. We proposed to solve this by modifying the up vector, or by in-
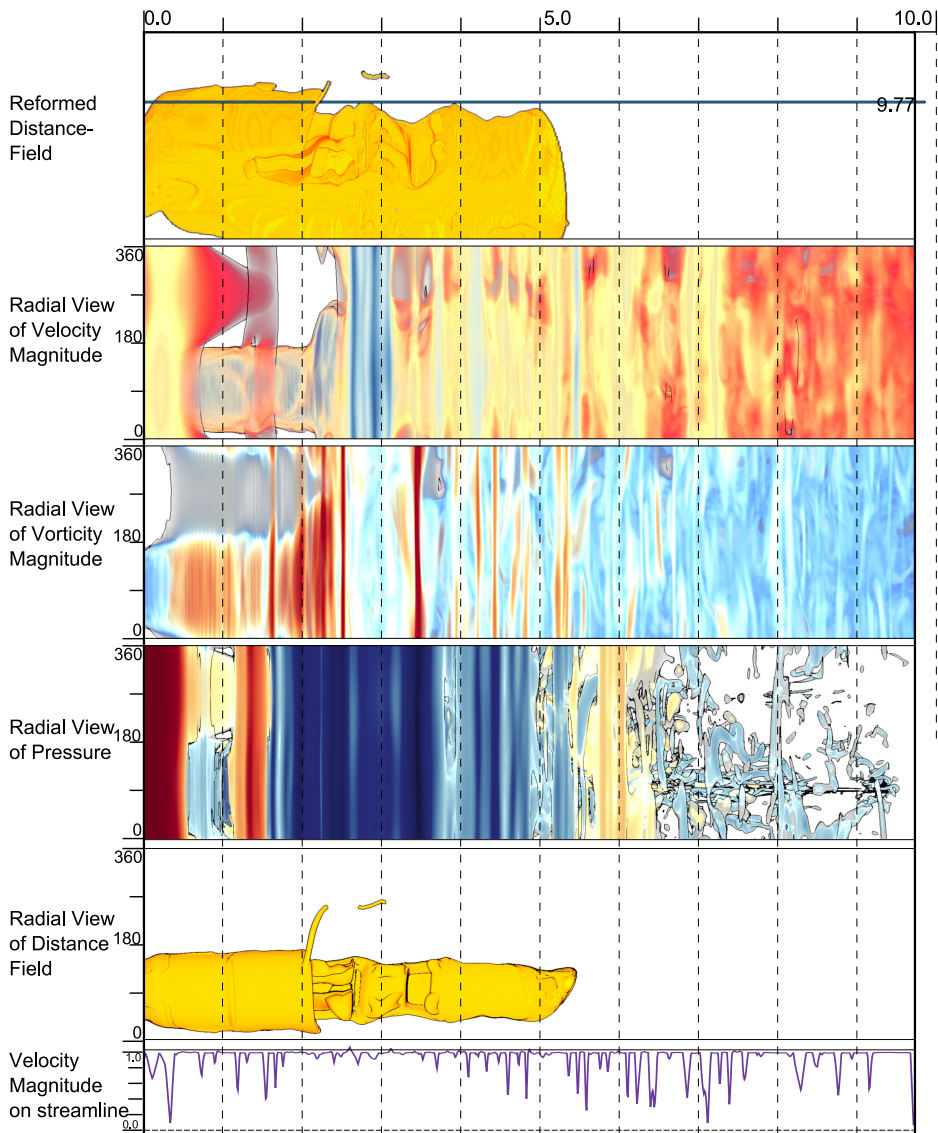
Figure 12: A plot showing streamline 5 as shown in Figure 11. The deformed car is shown on top, which then acts as a spatial reference for the measurements below. The three measurements below are radially ray-casted streamline-centric views, and below a graph tracing a value on the streamline.

creasing the smoothing factor, but the smoothing kernel could be expanded to take these longer sections into account, and smooth over them by using a varying smoothing factor.

In cooperation with our application partners we have used our prototype applications to show how curve-centric visualizations, combined with application specific data, can create effective and compelling comparative visualizations. Moreover, we have produced a more general approach, without the same application specific requirements as shown in previous techniques such as virtual colonoscopy or curve planar reformation, one that ideally could be used for numerous other applications as well.

6 Acknowledgments

Bibliography

[1] C. Ahlberg. Spotfire: an information exploration environment. *SIGMOD Record*, 25:25–29, 1996.

[2] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data: A systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[3] G. Andrienko, N. Andrienko, J. Dykes, S. Fabrikant, and M. Wachowicz. Geovisualization of dynamics, movement and change: key issues and developing approaches in visualization research. *Information Visualization*, 7(3):173–180, 2008.

[4] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data – A Systematic Approach.* Springer, 2006.

[5] N. Andrienko and G. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Trans. Visualization and Computer Graphics*, 2010. (RapidPost).

[6] A. Artero, M. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. *Proc. of IEEE Symp. on IN-FOVIS*, 2004.

[7] Asa data expo 2009. http://stat-computing.org/dataexpo/2009. [Online; accessed Nov-2010].

[8] Audacity. The Free Audio Editor and Recorder. audacity.sourceforge.net. [Online; accessed Nov-2010].

[9] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Trans. Visualization and Computer Graphics (Vis 2008)*, 14(6):1428–1435, 2008.

[10] R. Bade, S. Schlechtweg, and S. Miksch. Connecting time-oriented data and information to a coherent interactive visualization. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112, 2004.

[11] A. H. Barr. Global and local deformations of solid primitives. *Siggraph Comp. Graph.*, 18(3):21–30, 1984.

[12] A. V. Bartrolí, R. Wegenkittl, A. König, and E. Gröller. Nonlinear virtual colon unfolding. In *Proc. IEEE Vis.*, pages 411–420, 2001.

[13] L. Bergman, B. Rogowitz, and L. Treinish. A rule-based tool for assisting colormap selection. In *Proc. Visualization*, pages 118–125, 1995.

[14] L. Berry and T. Munzner. BinX: Dynamic exploration of time series datasets across aggregation levels. In *Proc. IEEE Symp. Information Visualization (InfoVis 2004)*, pages 215–216, 2004.

[15] Z. I. Botev. A novel nonparametric density estimator. *Postgrad. Sem. Series, Math. , The Univ. of Queensland*, 2006.

[16] B. W. Boyle, R. Madhavan, and J. Jundt. Wired pipe joint with current-loop inductive couplers, 2003. Patent No.: US 6641434.

[17] A. Braseth, V. Nurmilaukas, and J. Laarni. Realizing the information rich design for the loviisa nuclear power plant. *American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT), 6*, 2009.

[18] A. Braseth, Ø. Veland, and R. Welch. Information Rich Display Design. In *Proceedings of the Fourth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls and Human-Machine Interface Technologies, Columbus, Ohio*, 2004.

[19] C. Brewer. Color use guidelines for data representation. In *Proc. Section on Statistical Graphics*, pages 55–60, 1999.

[20] C. A. Brewer. Color advice for maps. http://ColorBrewer.org.

[21] M. D. Buhmann. *Radial basis functions*. Cambridge Uni. Press, 2003.

[22] M. Chen, C. Correa, S. Islam, M. W. Jones, P.-Y. Shen, D. Silver, S. J. Walton, and P. J. Willis. Manipulating, Deforming and Animating Sampled Object Representations. *Comp. Graph. For.*, 26(4):824–852, 2007.

[23] M. Chen, D. Silver, A. S. Winter, V. Singh, and N. Cornea. Spatial transfer functions: a unified approach to specifying deformation in volume modeling and animation. In *Proc. Vol. Graph.*, pages 35–44. ACM, 2003.

[24] R. Chin and C. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys (CSUR)*, 18(1):67–108, 1986.

[25] C. Correa, D. Silver, and M. Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE TVCG*, 12(5):1069–1076, 2006.

[26] R. Crawfis and N. Max. Texture splats for 3d scalar and vector field visualization. In *Proc. IEEE Visualization Conf. (Vis '93)*, pages 261–266, Oct 1993.

[27] O. Daae Lampe and H. Hauser. Curve density estimates. *Computer Graphics Forum*, 30(3):633–642, 2011.

[28] O. Daae Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis 2011)*, pages 171–178, March 2011.

[29] O. Daae Lampe, J. Kehrer, and H. Hauser. Visual analysis of multivariate movement data using interactive difference views. In *Proceedings of Vision, Modeling, and Visualization (VMV 2010)*, pages 315–322, 2010.

[30] M. desJardins and P. Rheingans. Visualization of high-dimensional model characteristics. In *Workshop on New Paradigms in Information Visualization and Manipulation*, pages 6–8, 1999.

[31] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. Eurographics/IEEE-TCVG Symp. on Visualization (VisSym 2003)*, pages 239–248, 2003.

[32] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Constructive theory of functions of several variables*, pages 85–100, 1977.

[33] S. G. Eick and G. J. Wills. High interaction graphics. *European Journal of Operations Research*, 81(3):445–459, 1995.

[34] G. Ellis and A. J. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE TVCG*, 13(6), 2007.

[35] M. Ericson. Keynote: Visualizing Data for the Masses: Information Graphics at The New York Times. *VisWeek*, 2007.

[36] Federal election commission. http://www.fec.gov/.

[37] J. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proc. of IEEE Symp. on INFOVIS*, 2002.

[38] D. Feng, L. Kwock, Y. Lee, and R. Ii. Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Transactions on Visualization and Computer Graphics*, 16:980–989, 2010.

[39] D. Fisher. Hotmap: Looking at geographic attention. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1184–1191, 2007.

[40] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics 7*, 1936. StatLib http://lib.stat.cmu.edu/.

[41] M. Florek and H. Hauser. Quantitative data visualization with interactive kde surfaces. In *Proceedings of the Spring Conference on Computer Graphics (SCCG 2010)*, pages –, May 2010.

[42] M. Florek and H. Hauser. Interactive bivariate mode trees for visual structure analysis. In *Proceedings of the Spring Conference on Computer Graphics (SCCG 2011)*, pages –, 2011.

[43] F. Frenet. Sur les courbes à double courbure. *Journal des Mathematiques Pures et Appliquees*, 17:437–447, 1852.

[44] R. Frigg and S. Hartmann. "models in science", the stanford encyclopedia of philosophy (summer 2009 edition), edward n. zalta (ed.). http://plato.stanford.edu/archives/sum2009/entries/models-science.

[45] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.

[46] J. Gain and D. Bechmann. A survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph.*, 27(4):1–21, 2008.

[47] V. D. Gesù and V. V. Starovoitov. Distance-based functions for image comparison. *Pattern Recognition Letters*, 20(2):207–214, 1999.

[48] A. Gray and A. Moore. Nonparametric density estimation: Toward computational tractability. In *SIAM Int. Conf. on Data Mining*, 2003.

[49] E. Gröller. Nonlinear ray tracing: Visualizing strange worlds. *The Visual Computer*, 11(5):263–274, 1995.

[50] E. Grundy, M. Jones, R. Laramee, R. Wilson, and E. Shepard. Visualisation of sensor data from animal movement. *Computer Graphics Forum*, 28(3):815–822, 2009.

[51] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Non-distorting flattening for virtual colonoscopy. In *MICCAI*, pages 358–366, 2000.

[52] M. Hao, D. Keim, U. Dayal, D. Oelke, and C. Tremblay. Density Displays for Data Stream Monitoring. *Computer Graphics Forum*, 27(3):895–902, 2008.

[53] M. Hao, D. Keim, U. Dayal, and T. Schreck. Multi-resolution techniques for visual exploration of large time-series data. In *EuroVis 2007*, pages 27–34, 2007.

[54] M. A. Harrower and C. A. Brewer. ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps. *The Cartograpic Journal*, 40(1):27–37, 2003.

[55] H. Hauser. Generalizing Focus+Context Visualization. In *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 305–327. Springer, 2005.

[56] S. He, R. Dai, B. Lu, C. Cao, H. Bai, and B. Jing. Medial axis reformation: A new visualization method for ct angiography. *Academic Radiology*, 8:726–733, 2001.

[57] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Proc. IEEE Symp. Information Visualization (InfoVis 2004)*, 3(1):1–18, 2004.

[58] W. Hong, X. Gu, F. Qiu, M. Jin, and A. Kaufman. Conformal virtual colon flattening. In *Symp. Solid Phys. Mod.*, pages 85–93. ACM, 2006.

[59] J. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[60] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: spreading aircraft trajectories across views to support iterative queries. *IEEE Trans. Visualization and Computer Graphics*, 15:1017–1024, 2009.

[61] Y. Jang, M. Weiler, M. Hopf, J. Huang, D. Ebert, K. Gaither, and T. Ertl. Interactively visualizing procedurally encoded scalar fields. In *Proc. of EG/IEEE TCVG Symp. on Vis. VisSym*, volume 4, 2004.

[62] T. Jankun-Kelly and K.-L. Ma. Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Trans. Visualization and Computer Graphics*, 7(3):275–287, 2001.

[63] F. Janoos, S. Singh, O. Irfanoglu, R. Machiraju, and R. Parent. Activity analysis using spatio-temporal trajectory volumes in surveillance applications. In *Proc. IEEE VAST*, pages 3–10, 2007.

[64] D. Jerding and J. Stasko. The information mural: A technique for displaying and navigating large information spaces. *Proc. IEEE Visualization Conf. (Vis 2002)*, 4(3):257–271, 2002.

[65] M. Jern and J. Franzén. GeoAnalytics—exploring spatio-temporal and multivariate data. In *Proc. Int'l. Conf. Information Visualization (IV '06)*, pages 25–31, 2006.

[66] J. Johansson, P. Ljung, and M. Cooper. Depth cues and density in temporal parallel coordinates. *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization*, 7:35–42, 2007.

[67] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[68] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, and M. E. Gröller. CPR - Curved Planar Reformation. *Proc. IEEE Vis.*, 0:37–44, 2002.

[69] A. Kanitsar, R. Wegenkittl, D. Fleischmann, and M. Gröller. Advanced curved planar reformation: flattening of vascular structures. *Proc. IEEE Vis.*, pages 43–50, Oct. 2003.

[70] J. Kehrer, P. Filzmoser, and H. Hauser. Brushing moments in interactive visual analysis. *Computer Graphics Forum*, 29(3):813–822, 2010.

[71] D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. *Visual Data Mining*, pages 76–90, 2008.

[72] D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proc. Int'l. Conf. Information Visualization (IV '06)*, pages 9–16, 2006.

[73] P. Kidwell, G. Lebanon, and W. Cleveland. Visualizing Incomplete and Partially Ranked Data. *IEEE TVCG*, 14(6), 2008.

[74] R. Kincaid. SignalLens: Focus+context applied to electronic time series. *IEEE Trans. Visualization and Computer Graphics (Vis 2010)*, 16(6):900–907, 2010.

[75] F. Klok. Two moving coordinate frames for sweeping along a 3d trajectory. *Computer Aided Geometric Design*, 3(3):217 – 229, 1986.

[76] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proc. IEEE Visualization Conf. (Vis 2003)*, pages 497–504, 2003.

[77] R. Kosara, F. Bendix, and H. Hauser. Timehistograms for large, time-dependent data. *Joint Eurographics–IEEE TCVG Symposium on Visualization*, 2004.

[78] Y. Kurzion and R. Yagel. Space deformation using ray deflectors. In *6th Eurographics Workshop on Rendering 95*, pages 21–32, 1995.

[79] N. Lee and M. Rasch. Tangential curved planar reformation for topological and orientation invariant visualization of vascular trees. *IEEE Eng. in Med. and Bio. Soc.*, pages 1073–1076, 2006.

[80] Z. Liu and J. Stasko. Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective. *IEEE Trans. Visualization and Computer Graphics*, 16(6):999–1008, 2010.

[81] H. Löffelmann and E. Gröller. Ray Tracing with Extended Cameras. *Journal of Visualization and Computer Animation*, 7(4):211–228, 1996.

[82] H. Löffelmann, T. Kučera, and E. Gröller. Visualizing Poincaré maps together with the underlying flow. In *Mathematical Visualization - Algorithms, Applications and numerics*, pages 315–328. Springer, 1998.

[83] K. Matkovic, W. Freiler, D. Gracanin, and H. Hauser. Comvis: a coordinated multiple views system for prototyping new visualization technology. In *Proceedings of the 12th International Conference Information Visualisation*, 7 2008.

[84] S. Miksch, A. Seyfang, W. Horn, and C. Popow. Abstracting steady qualitative descriptions over time from noisy, high-frequency data. *AIMDM '99 Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pages 281–290, 1999.

[85] M. Minnotte, D. Marchette, and E. Wegman. New terrain in the mode forest. *COMPUTING SCIENCE AND STATISTICS*, pages 473–477, 1998.

[86] M. C. Minnotte and D. W. Scott. The mode tree: a tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, 2, 1993.

[87] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A Four-level Focus+Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data. *Computer Graphics Forum*, 27(3):775–782, 2008.

[88] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

[89] Norwegian Meteorological Institute. eKlima. **eklima.met.no**. [Online; accessed Nov-2010].

[90] M. Novotný and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE TVCG*, 12(5), 2006.

[91] G. Nygaard. elad. http://goo.gl/OTkXK. Accessed 01.09.2011.

[92] H. Pagendarm and F. Post. *Comparative Visualization: Approaches and Examples*. Delft University of Technology, Faculty of Technical Mathematics and Informatics, 1995.

[93] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3), 1962.

[94] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum*, 22:775–792, 2003.

[95] E. Ramos and D. Donoho. 1983 ASA data exposition dataset. http://lib.stat.cmu.edu/datasets/.

[96] J. Rasmussen. Skills, rules, knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, 13:257–266, 1983.

[97] J. Rasmussen. The role of hierarchical knowledge representation in decisionmaking and system management. *IEEE Transactions on Systems, Man, & Cybernetics*, 15(2):234–243, 1985.

[98] H. Reijner. The development of the horizon graph. In *IEEE Visualization Workshop: From Theory to Practice: Design, Vision and Visualization*, 2008.

[99] C. Rezk-Salama, M. Scheuering, G. Soza, and G. Greiner. Fast volumetric deformation on general purpose hardware. In *Proc. on Workshop on Graph. Hardware*, pages 17–24. ACM, 2001.

[100] P. Rheingans and M. desJardins. Visualizing high-dimensional predicitive model quality. In *IEEE Visualization*, pages 493–496, 2000.

[101] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3), 1956.

[102] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. *Proc. IEEE Symp. Information Visualization (InfoVis 2005)*, pages 173–180, 2005.

[103] R. Scheepens, N. Willems, H. van de Wetering, , and J. J. van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis 2011)*, pages 147–15, March 2011.

[104] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization.* Wiley-Interscience, illustrated edition, 1992.

[105] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proc. Siggraph '86*, pages 151–160. ACM, 1986.

[106] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages*, pages 336–343, 1996.

[107] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *CHI '08: Proc. of SIGCHI on Human factors in computing systems*, pages 1237–1246, 2008.

[108] B. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, 1986.

[109] W. Silvert. Modelling as a discipline. *International Journal of General Systems*, 30(3):261–282, 2001.

[110] K. Singh and E. Fiume. Wires: a geometric deformation technique. In *Proc. Comp. Graph. and Interactive Tech.*, pages 405–414. ACM, 1998.

[111] J. P. Snyder. *Flattening the Earth: Two Thousand Years of Map Projections.* University of Chicago Press, 1993.

[112] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Visualization and Computer Graphics*, 8(1):52–65, 2002.

[113] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3):80, 2007.

[114] W. Szewczyk. Streaming data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(1):22–29, 2011.

[115] Tableau. http://www.tableausoftware.com/. Accessed 01.09.2011.

[116] D. Tarn. An introduction to kernel density estimation. http://school.maths.uwa.edu.au/~duongt/seminars/intro2kde/, 2001.

[117] J. Thomas and K. Cook. *Illuminating the Path: Research and Development Agenda for Visual Analytics.* IEEE-Press, 2005.

[118] T. J. True and J. F. Hughes. Volume warping. In *Proc. IEEE Vis.*, pages 308–315, 1992.

[119] E. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative.* Graphics Press, 1997.

[120] E. Tufte. *Beautiful evidence*, volume 23. Graphics Press Cheshire, CT, 2006.

[121] E. R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, 1983.

[122] E. R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative.* Graphics Press, 1997.

[123] J. W. Tukey. *Exploratory Data Analysis.* Addison-Wesley, 1977.

[124] B. A. Turlach. Bandwidth Selection in Kernel Density Estimation: A Review. In *CORE and Institut de Statistique*, 1993.

[125] L. A. Tweedie, R. Spence, D. Williams, and R. Bhogal. The attribute explorer. In *In Proc. of the Video Track of the ACM Conference on Human Factors in Computing Systems*, pages 435–436, 1994.

[126] V. Verma and A. Pang. Comparative flow visualization. *IEEE Trans. Visualization and Computer Graphics*, 10(6):609–624, 2004.

[127] K. Vicente and J. Rasmussen. Ecological interface design: theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, jul/aug 1992.

[128] T. Vrtovec, B. Likar, and F. Pernus. Automated curved planar reformation of 3D spine images. *Physics in Medicine and Biology*, 50(19):4527, 2005.

[129] D. F. Walnut. *An Introduction to Wavelet Analysis.* Springer, 2004.

[130] M. Wand and M. Jones. *Kernel Smoothing.* Monographs on Statistics and Applied Probability 60. Chapman & Hall, 1995.

[131] G. Wang, G. McFarland, B. Brown, and M. Vannier. GI tract unraveling with curved cross sections. *IEEE Trans. Med. Img.*, 17(2):318–322, 1998.

[132] M. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *Proc. IEEE Visualization Conf. (Vis '94)*, pages 326–336, 1994.

[133] R. Westermann and C. Rezk-Salama. Real-time volume deformations. *Comp. Graph. Forum*, 20(3):443–451, 2001.

[134] U. Weyer, A. Braseth, M. Eikås, L. Hurlen, and P. Kristiansen. Safety presentation in large screen displays - a new approach. *SPE Intelligent Energy Conference and Exhibition*, 2010.

[135] G. Whittaker and D. Scott. Nonparametric regression for analysis of complex surveys and geographic visualization. *Sankhyā: The Indian Journal of Statistics, Series B*, 1999.

[136] N. Willems, H. van de Wetering, and J. van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[137] D. Williams, S. Grimm, E. Coto, A. Roudsari, and H. Hatzakis. Volumetric curved planar reformation for virtual endoscopy. *IEEE TVCG*, 14(1):109–119, 2008.

[138] M. Wohlfart and H. Hauser. Story Telling for Presentation in Volume Visualization. In *EuroVis*, 2007.

[139] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas. Dynamic visualization of transient data streams. *Information Visualization, IEEE Symposium on*, pages 97–104, 2003.

[140] P. C. Wong, H. Foote, D. L. Kao, L. R. Leung, and J. Thomas. Multivariate visualization with data fusion. *Information Visualization*, 1, 2002.

[141] J. Woodring and H.-W. Shen. Multi-variate, time-varying, and comparative visualization with contextual cues. *IEEE Trans. Visualization and Computer Graphics*, 12:909–916, 2006.

[142] D. Yang, Z. Xie, E. A. Rundensteiner, and M. O. Ward. Managing discoveries in the visual analytics proc. *SIGKDD Explor. Newsl.*, 9(2), 2007.

[143] J. S. Yi, Y. A. Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[144] S. Zambal, A. Schöllhuber, K. Bühler, and J. Hladuvka. Fast and robust localization of the heart in cardiac MRI series. *Proc. of Int. Conf. on Computer Vision Theory and Applications*, 2008.

[145] A. Zhou, Z. Cai, L. Wei, and W. Qian. M-kernel merging: Towards density estimation over data streams. *IEEE Proceedings of the Eighth International Conference on Database Systems for Advanced Applications (DASFAA'03)*, 2003.