Paper B

Curve Density Estimates

Ove Daae Lampe[1,2], and Helwig Hauser[1]

[1]Department of Informatics, University of Bergen, Norway
[2]Christian Michelsen Research, Norway

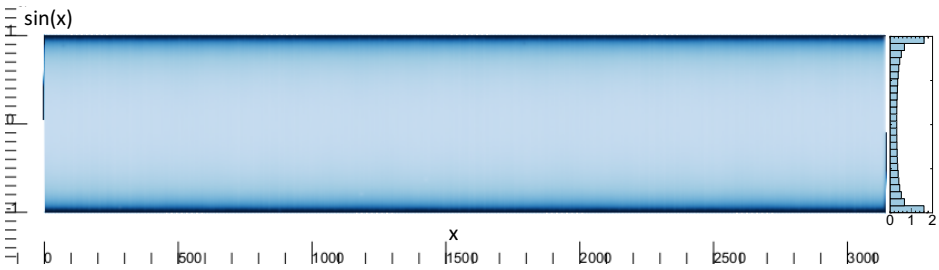

Figure 1: The Curve Density Estimate of a high frequency sine curve with a normalized histogram of evaluated values densely sampled along the x axis. Our continuous representation of this curve closely matches that of the histogram.

Abstract

In this work, we present a technique based on kernel density estimation for rendering smooth curves. With this approach, we produce uncluttered and expressive pictures, revealing frequency information about one, or, multiple curves, independent of the level of detail in the data, the zoom level, and the screen resolution. With this technique the visual representation scales seamlessly from an exact line drawing, (for low-frequency/low-complexity curves) to a probability density estimate for more intricate situations. This scale-independence facilitates displays based on non-linear time, enabling high-resolution

This article was published in *Proceedings of Eurographics/IEEE-VGTC Symp. on Visualization (EuroVis 2011)*, 30(3), pages 633–642, 2011, and presented at EuroVis in Bergen, Norway by Ove Daae Lampe.

accuracy of recent values, accompanied by long historical series for context. We demonstrate the functionality of this approach in the context of prediction scenarios and in the context of streaming data.

1 Introduction

In the context of time-dependent data the drawing of function graphs is one of the most natural and at the same time one of the most effective data visualization techniques. As long as the spatial complexity of the graph is limited, this immediate translation of data into a graph is straight forward and provides intuitive results. If the curve to draw, however, becomes very long or the spatial complexity increases, for example when considering a fractal curve, then the simple plotting of such a curve or graph will likely result in problems with overdraw and cluttering. This overdraw in an example graph led us to the question: Why did our regular graph of a sine curve look so different when its samples where drawn in a scatterplot instead? (see Figure 2 a and c). The scatterplot, when drawn with transparency, resembles the histogram of these values, as shown in Figure 3. This histogram shows the distribution of the evaluated values, but the curve representation completely obscures this distribution, even when applying transparency, as shown in Figure 2b. In this paper, we investigate an alternative way of rendering such curves, that does not display the same problems, but keep the clear benefits of the regular curve.

As, perhaps, a very trivial summary, a function graph displays a single measured value, on one axis, with its continuous changes over another axis. Unless the changes are piecewise continuous, a curve is not an appropriate choice of visualization.

One of the biggest challenges when drawing function graphs is that they are mainly useful for displaying frequencies that, on the extreme, is at least greater than the pixel with of the display. The common way to deal with this is to either constrain/zoom in on the axis, or to aggregate values, to "smooth" out rapid changes. In Figure 4, aggregated stock prices for Intel are shown, with the black curve being the center-shifted moving mean, enveloped in the curves' standard deviation. The outer polygon is the moving max-min. This enveloped curve, as described by Miksch et al. [84], captures the overall movement of the underlying data very well, and its standard deviation polygon reveals important information about the frequency or stability of the smoothed data. This method works best when the data is close to the normal distribution (or at least unimodal).In the case of a bi- or multimodal data distribution, however, this aggregation looses certain expressiveness. The visualized mean can easily associate with highly improbably data values, for example, in the middle between two modes. Moreover, the moving mean relies on a certain window, which either provides lagging results or it is undefined for the latest values.

As a simple example we consider a sine curve from zero to a number larger than the amount of available pixels in the horizontal direction. A naïve approach to display this curve is shown in Figure 2a, which suffers from overdraw, and would mainly only display the extent of the curve. A first approach on how to solve this problem could be to apply transparency, shown in Figure 2b. The transparency

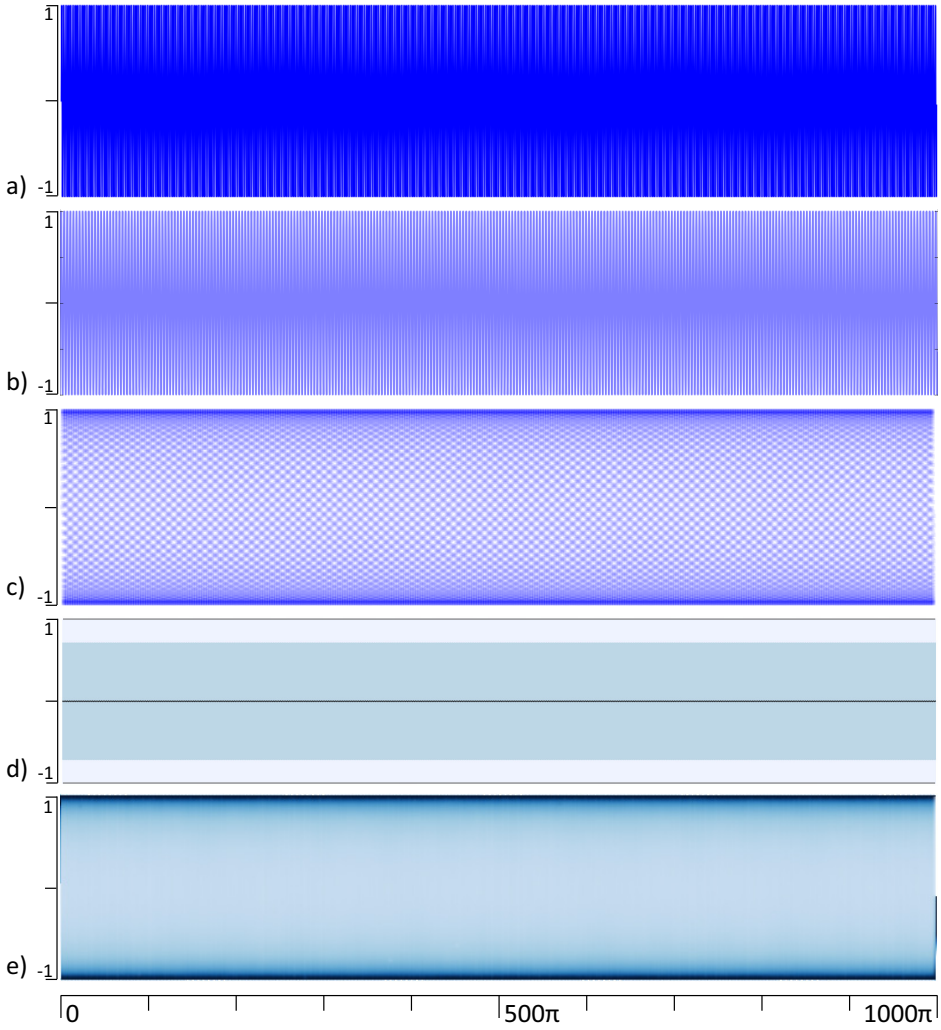

Figure 2: Figures displaying the sine curve from zero to 1000π. In the top figure, a, an opaque line is used, and because of overdraw, displays only the extent of the function. In the second figure, b, a transparent line is used. The third figure, c, is a scatter-plot of the samples drawn transparent, and shows the same distribution as the histogram. The fourth figure, d, is aggregated with moving mean, standard deviation and extent. As opposed to Figure 4, this data is unsuitable for this type of aggregation. In the bottom figure, our technique, the Curve Density Estimate, is applied, and the distribution corresponds with that found in the histogram in Figure 3.

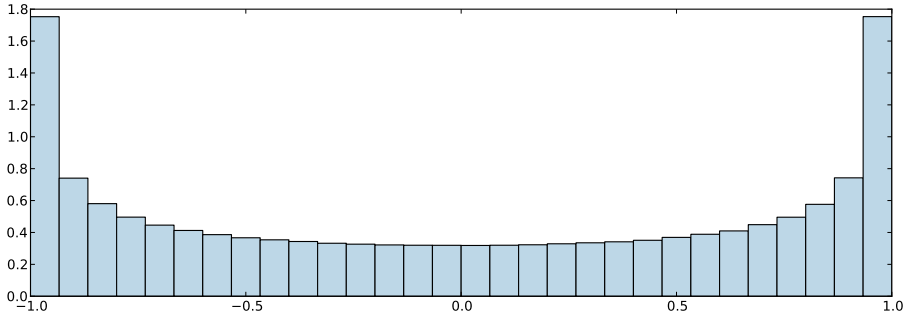

Figure 3: 30 bins histogram of $y = \sin(x)$ for regularly sampled values of x.

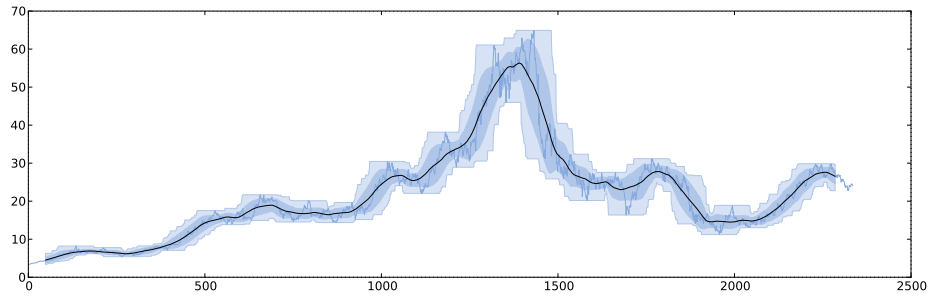

Figure 4: Intel opening stock price with a moving average, standard deviation and extent.

could correctly display the amount of overdraw, but this does not correctly display the distribution of the curve. The distribution of a curve, alternatively described as its continuous histogram by Bachthaler and Weiskopf [9], is found by taking regular samples along its parameter axis, and inserting evaluated values in a histogram. The histogram for the sine curve is shown in Figure 3. It is worth noting that the curve with transparency will have a single visible mode at zero, which is almost the opposite of what the histogram indicates, with two modes at one and minus one.

A second technique on how to deal with this high frequency sine curve could be to first aggregate it. In Figure 2d, we show, similar to Figure 4, the moving average, standard deviation, and extent. Using a sufficiently large window, the average of the sine curve is stable at zero, and its standard deviation is constant. Part of the problem is that this way a model, with a normal distribution, is enforced onto the data. If there is a mismatch between the assumed model and the data, such a visualization will not be expressive.

With our technique applied to the sine curve, as shown in Figure 1 and Figure 2e, we recreate the distribution without any prior knowledge of model, and

will do so independent of frequency, zoom level and screen resolution. That said, we do not propose to replace the aggregation techniques, as in Figure 4 or other techniques, where the model is known, but provide a default view, that can either be used before the model is established (if it exist), or to investigate how well a selected model fits the data.

With this paper we introduce a novel way of displaying function graphs, that also supports:

- Graphs with a frequency higher than the pixel-width of its display
- Smooth transition between high frequency areas and single line curve
- The creation of a probability density estimate that does not assume a normal distribution of values
- The probability density of both single and multiple curves (and a mix between those)

In the following, we first discuss related work, then move on to the theoretical details of curve density estimates, before we add technical implementation details. Lastly, we apply our technique to real world data before providing the summary and conclusions.

2 Related Work

Existing techniques, improving or extending the curve, fall briefly into the categories: compact views, overdraw views and, distribution views.

Compact Views: By utilizing techniques to compress the value axis, the aspect ratio is improved such that longer time-series can be shown on less space without contracting the time axis, and thus avoid the frequency problem. Saito et al. [102] designed a compact graph view, that utilizes a colored banding to overlay multiple ranges of the curve on top of each other. Using this banding, which was further refined into the horizon graph by Panopticon [98], a precise value can be read out, while reducing the physical height down to an eighth. When the value range of a process is known, and also can be defined in levels, such as low, normal and high, all values in these ranges can be replaced with colors to produce a compact visualization, as described by Bade et al. [10]. They provide an interesting example of body temperature graphs, where there are clearly defined normal levels. Another compact graph visualization are the Sparklines as introduced by Tufte [120], which strips the curve down to a text-line sized graph, that can even be included mid-text. As a separate thread of compact visualization techniques, is the pixel based category, where each sample is displayed using a colored pixel. In 2008, Hao et al. [52] provided an evaluation on how best to place such pixels, while keeping temporal coherence. While not quite a compact view, Kincaid proposed combining high frequency time series with a

focus+context interaction [74]. This interction provides both an overview, and a detailed view down to the individual samples.

Overdraw Views: Techniques that deal with visual clutter of high frequency by introducing schemes to blend multiple overdraws. Most visualization packages allows the user to specify different opacities, effectively implementing an overdraw view. In 2002, Jerding and Stasko introduced the Information Mural [64] to deal with high frequency graphs and other cluttered 2D visualizations. This technique downscales large, original, and uncluttered views to miniatures, while counting the overdraws to each pixel. This overdraw count is then used to apply a greyscale color.

Distribution Views: Techniques that by aggregation deduce the distribution of a single, or multiple curves. Hochheiser and Shneiderman [57] utilized envelopes that displayed the full extent of curves, in their TimeSearcher application. In 2004, Kosara et al. [77] described the TimeHistogram, where the time axis was divided into intervals, and a separate histogram was calculated for each of these intervals. These histograms, with colored 1D representations, was then in turn displayed along the time axis. In the work by Muigg et al. [87] multiple curves where binned while aggregating both the count and the directions, to create a visualization close to that of a flow field, utilizing line integral convolution (LIC) to overlay direction on top of the frequency. Bade et al. [10] introduced an extension to the information mural [64], that adds the median, the 25 and 75 percentiles and extent. Which is similar to BinX [14] which visualizes long time series by binning along the time axis at different levels of aggregation and then displays mean, minimum, maximum value, and standard deviation per bin. Johansson et al. [66] discussed a blending scheme to introduce temporal changes in parallel coordinate plots (PCP). Their implementation aggregates continuous changes on top of each other forming, what can be described as a continuous 1D histogram from discrete samples. This 1D histogram is then the basis for creating a polygon that is drawn to the next axis in the PCP. Feng et al. [38] also introduced an extension of PCP, that is the result of mapping the 2D KDE between each axis, into its corresponding parallel coordinate version. Furthermore, Feng et al. [38] also introduced several enhancements to interaction and brushing techniques to better suite frequency data.

The technique proposed in this paper is an extension of our previous work [28], where we first introduced the concept of a line kernel to kernel density estimation. The line kernel is used to reconstruct continuous changes, by connecting consecutive samples, forming an elongated kernel, and that integrates up to one independent of the distance between samples. In this work we extend this line kernel, and show how to reduce it into an exact and continuous, parametric formulation. In our previous work, we utilized a table of pre-integrated convolved results, whereas this extension allow us to directly evaluate the exact result. Additionally, in this work, we introduce a curve visualization, called *curve density estimates* (CDE), that provide distributional characteristics along the time axis,

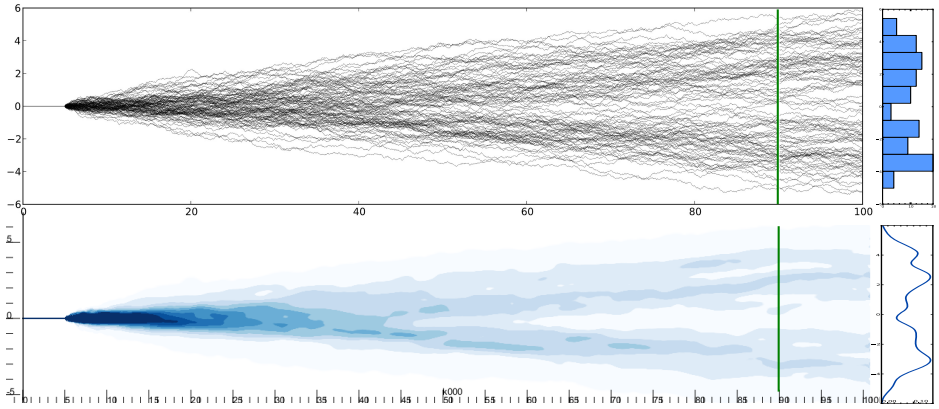

Figure 5: 100 cumulative random curves with a slight bimodal trend. The top graph show the curves with slight transparency. All the samples at the green line, at $x = 90$, are drawn using a histogram and a 1D KDE in to the right. The graph to the bottom shows the CDE. Note how the 1D KDE corresponds to the green line drawn over the CDE as well.

comparable to the concept of a continuous 1D KDE. This visualization is enabled by a moving column based normalization scheme further detailed in Section. 3. Several other extensions are also provided here, over our previous work, e.g., non-linear time, single curve to multiple curves transition.

3 Curve Density Estimates (CDE)

The main rationale behind the use of kernel density estimation (KDE) as a basis is that it does not impose any model on the data. Given a discrete set of samples, with an appropriate bandwidth and kernel, KDE can truthfully approximate any probability density estimate (PDE). For an extensive overview of KDE we refer to Silverman [108]. The KDE is defined as the sum of a number of kernels, one kernel per sample. With $(x_1, x_2, ..., x_n)$ being samples corresponding to an unknown density f, the according KDE is defined as

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) \quad = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big), \tag{1}$$

with K a suitable kernel. As the kernel, often the normal distribution, $N(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\ e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, is used, with μ being the mean, σ^2 the variance, and h the bandwidth, or kernel size.

The blue vertical graph to the bottom right of Figure 5 shows an 1D KDE. This KDE is created from the set of points where the black curves intersect the

green line in the upper-left graph in the same figure. This 1D KDE clearly reveals the bimodal nature of this dataset. Our curve density estimates (CDE) in the lower graph in Figure 5, can be interpreted as a continuous series of these 1D KDEs. However, instead of modeling our solution by expanding 1D KDEs, we find a solution via a 2D KDE. A standard 2D KDE can be created using the unconnected sample-points from the dataset, inserted into Eq. 1. This approach will not create a continuous distribution when the samples get far apart, but rather be the distribution of the previously mentioned scatterplot (2c). The consecutive samples in the time series represent a continuous change from one value to the next, and thus the probability, given two samples, should not be 0.5 at each sample, but rather be distributed evenly from one to the next. We achieve this by building upon a line kernel L_k defined by two consecutive data samples, and their positions $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ [28]:

$$L_k(\mathbf{x}) = \int_0^1 c_i K_{\mathbf{H}}\big(\mathbf{x} - ((1-\phi)\mathbf{p}_i + \phi\mathbf{p}_{i+1})\big)\,\mathrm{d}\phi, \tag{2}$$

with $K_{\mathbf{H}}$ being the 2D normal distribution kernel. To enable a proper reconstruction from uneven sampling in time, we insert the elapsed time between the two samples in the scaling factor c_i.

We now can reduce Eq. 2 to 1D by only considering values on the line defined by $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$. We name this 1D equation $L_{k1D}(x)$. Furthermore we define the 2D points $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ to their 1D equivalents (they are per definition *on* this line), q_i and q_{i+1}, respectively. This 1D line kernel is then defined as the integral of Gaussians placed along a line segment. So for any point x, we observe that $L_{k1D}(x)$ is defined by the sum of these kernels, and that all those kernels incrementally have a mean/μ that is greater and greater than x. By turning this problem around, we deduce that the integral on one position of kernels with its mean moving away, is equal to the finite integral over a single kernel. The integral of the normal distribution is a cumulative distribution function (cdf). This distribution function is defined by

$$\mathrm{cdf}(\mathrm{x}, \mu, \sigma) = \frac{1}{2}\left(1 + \mathrm{erf}\big(\frac{x-\mu}{\sqrt{2\sigma^2}}\big)\right). \tag{3}$$

Considering a point x where $x < q_1$, and for explanation purposes a $q_2 \to \infty$. At this point x, $f(x) = \mathrm{cdf}(\mathrm{x}, \mathrm{q}_1, \sigma)$, since it is equal to the unbound integral of all the kernels starting from q_1 towards ∞. However, since q_2 is actually a finite value and we do not have any contribution from kernels beyond this point, we have to remove this from our equation. The contribution from all kernels starting at q_2 going towards ∞ is similarly, $f(x) = \mathrm{cdf}(\mathrm{x}, \mathrm{q}_2, \sigma)$. We then conclude, that for the two points q_1 and q_2, where $q_1 < q_2$, the line kernel, in 1D, is given by:

$$L_{k1D}(x) = \frac{1}{|q_2 - q_1|}\big(\mathrm{cdf}(\mathrm{x}, \mathrm{q}_1, \sigma) - \mathrm{cdf}(\mathrm{x}, \mathrm{q}_2, \sigma)\big), \tag{4}$$

with $|q_2 - q_1|$, the length between these points, applied for normalization, since

$$\int \mathrm{cdf(x, q_1, \sigma)} - \mathrm{cdf(x, q_2, \sigma)} dx = q_2 - q_1. \tag{5}$$

One important quality of this line kernel is when q_1 approaches q_2

$$\lim_{q_1 \to q_2} L_{k1D}(x) = N(x) \tag{6}$$

the line kernel approaches the normal distribution, $N(x)$, an observation which was also previously made by Kniss et al. [76].

The next step, is to expand this 1D line kernel, over to our 2D case again. In our previous work [28], we relied on the product kernel to define the line kernel. Here, to expand our 1D parametric line kernel to 2D, we also rely on a product kernel, but we let the first kernel be our 1D line kernel, and the other, the normal distribution. Let $\mathbf{w}$ be the point $\mathbf{x}$ projected onto the line L defined by $\mathbf{p}_1$ and $\mathbf{p}_2$, i.e., $\mathbf{w} = \mathbf{x} + |\mathbf{r} \cdot (\mathbf{x} - \mathbf{p}_1)|\mathbf{r}$, with $\mathbf{r}$ a unit vector perpendicular to the line. Then let u be the distance $|\mathbf{p}_1 - \mathbf{w}|$ and v the distance $|\mathbf{x} - \mathbf{w}|$. This then gives the new and parametric 2D definition of the line kernel:

$$L_k(\mathbf{x}) = c_i L_{k1D}(u) \cdot N(v) \tag{7}$$

The KDE using our line kernel that will continuously reconstruct the sample points $(\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n)$ is defined by

$$f_{Lk}(\mathbf{x}) = \sum_{i=0}^{n-1} L_k(\mathbf{x}, \mathbf{p}_i, \mathbf{p}_{i+1}) \tag{8}$$

By evaluating this KDE we get a density field with the integral

$$\int f_{Lk}(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{n} c_i, \tag{9}$$

since individually, all kernels integrate up to one, but are scaled by c_i. Usually, to create a probability density estimate, we would normalize by this integral, but instead we propose to normalize it after rasterization, and then only column by column, individually. Given the 2D grid, G, evaluated by $f_{Lk}(\mathbf{x})$, we create a column-normalized grid G_n by,

$$G_n[i, j] = G[i, j] / \sum_{\hat{j}=0}^{h} G[i, \hat{j}], \tag{10}$$

where h denotes the height of the grid. Figure 6a, displays two curves, that coincide before separating, and Figure 6b the rasterized result, after this column-wise

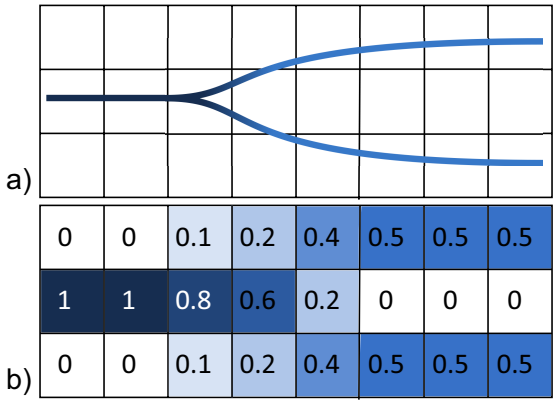

Figure 6: Two coinciding curves splitting up, in a, and the rasterized result after normalization, in b. Note that all columns sum up to one.

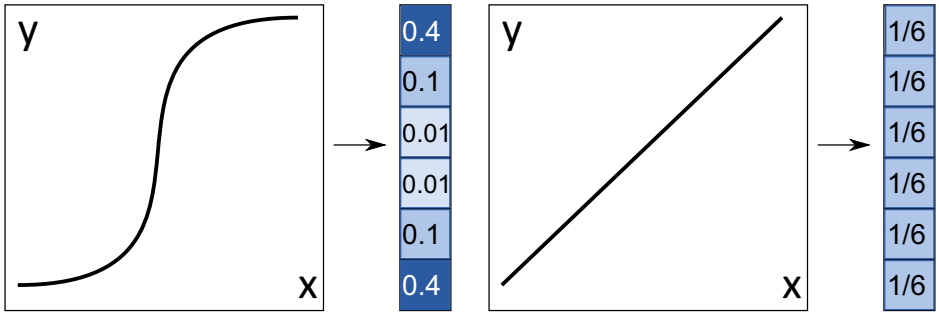

Figure 7: Two different curves, on the left, and their corresponding views, on right, after being rescaled down to one column.

normalization. The rationale for this normalization is to, first, have an intuitive number indicating how much time the curves spent where, and secondly be able to interpret every column as a 1D probability density estimate. A one indicating that all the curves was here 100% of the time, and 0.5, 50%, regardless of how many curves are used, and importantly, if non-linear time is used, regardless of how compressed time is. Utilizing this normalization will render single curves, with small changes, with the most intense values from the chosen color map, and also effectively applying anti-aliasing. However, when the curve reside in the same column, with large changes, e.g., when zooming out, or showing long time-series, the normalization will give the probability density/continuous histogram of where the curve "spent its time". Figure 7 shows two different curves, and their rasterized results after rendering them into a single column.

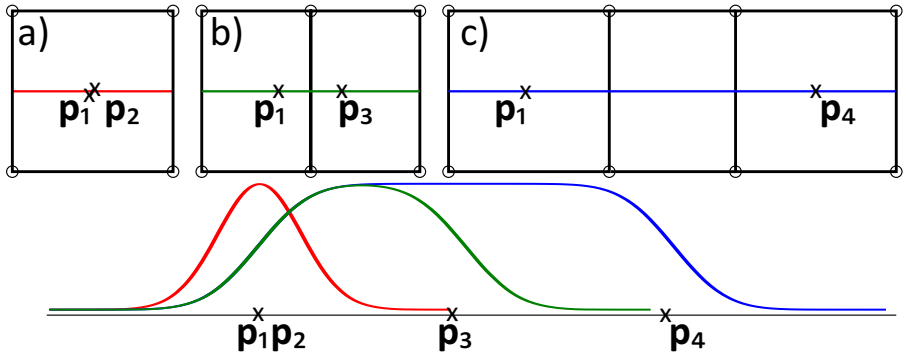

Figure 8: The three different proxy geometry schemes for reconstructing line kernels. The circles indicate the vertices needed. The colored curves below, depicts the evaluated kernel in the corresponding cross-sections above (not normalized).

4 Technical Details

While we see the usefulness of this technique in off-line renderings for static displays, we will, in this section, first address the challenges we face when either rendering real-time streaming data, or need interactivity, e.g. zooming and panning. In order to overcome the challenges and achieve good frame-rates, we offload all the calculation to the GPU. When rendering large time-series, we store vertex arrays, containing the samples, in GPU residing memory, in chunks sorted temporally. Because of this temporal ordering, we perform an intersection test, with the chunks' span vs. that of the view, to determine if it should be rendered or not. The chunk containing the most recent values, however, are stored in main memory, to be able to constantly append new values to it. To minimize the memory usage, the memory structure only contains the samples once, and the proxy geometry, needed by the line kernels, is constructed in a geometry shader step. The geometry shaders input are the consecutive samples, using the GL_LINE_STRIP_ADJACENCY, and it will output the proxy geometry in three distinct ways.

In the first case, a single, unconnected kernel is constructed as a quad. This case is used when the two consecutive samples' distance is less than a threshold $|\mathbf{p}_i - \mathbf{p}_{i+1}| \leq \epsilon$, and that threshold, expressed in terms of pixels, should be one. In our experience, however, there is no significant visual change when increasing ϵ to three pixels (given that the bandwidth is larger than that as well). This case is depicted in Figure 8a.

In the second case, when $\epsilon < |\mathbf{p}_i - \mathbf{p}_{i+1}| \leq c\sigma^2$, we have a line kernel, but one that have a single mode, or maxima. This occurs for $c \approx 5$ (see our previous

work[28] for a discussion on this clamping, and for the errors introduced), when distance is approx. five times that of the bandwidth, since the line kernel L_k is in effect the sum of multiple normal distributions, and its edge will follow the *cdf* function. This line kernel consists of two quads, meeting at the center-point between the points $(\mathbf{p}_i + \mathbf{p}_{i+1})/2$. This case is depicted in Figure 8b.

The third case, is when the distance between the points is sufficiently large, i.e., $|\mathbf{p}_i - \mathbf{p}_{i+1}| > c\sigma^2$, to have a "flat" region between them. In this case the proxy geometry consists of three quads, two for the end caps, and one for the continuous region in between. This case is depicted in Figure 8c. In this case, we can simplify the calculation of the line kernel L_k, to only include the normal kernel perpendicular to the centerline, for the middle segment.

Another usage for the geometry shader, in addition to the three previous cases, is applied when non-linear axes are used. The geometry shader evaluates, according to the distance between the two points, the error introduced by a single linear kernel, and performs a subdivision if needed.

Now, after the geometry is constructed, the kernels evaluate Eq. 7, and their sum is stored in a 32bit floating point texture. The fragment shader, parameterized with u and v, respectively, along and across the geometry, evaluating Eq. 7, calculates $N(u)$ using a table lookup, and the $L_{k1D}(v)$ using existing erf hardware accelerated GLSL implementation. Then, two steps remain, namely, the column normalization, and the application of a color-map. Before the normalization of the columns, however, we first need to calculate the sum per column. We store these sums in a 1D floating point texture, with the same width as the source image, the 2D rasterized grid. To calculate the sum, we first bind the 1D texture as a frame buffered object (FBO), as the render target, and then in a fragment shader, iterate over all texels in the corresponding column of the 2D texture, calculating the sum. As the last step, we apply the normalization division per fragment, while simultaneously applying the color-map.

5 Applications

In this section we cover a varied set of applications for the technique of curve density estimates (CDE). With these applications we show both the generality of the proposed technique while also highlighting specific areas where this technique can outperform the current state of art. Some of these examples are also covered as videos in the supplemental material to demonstrate the interactivity.

5.1 High Frequency Curves

A high frequency curve has significant amplitude fluctuations within short spans, in terms of the visualized area. Sound is an excellent example for high frequency curves. In Figure 9 we display the CDE of the waveform from Beethovens Sym-

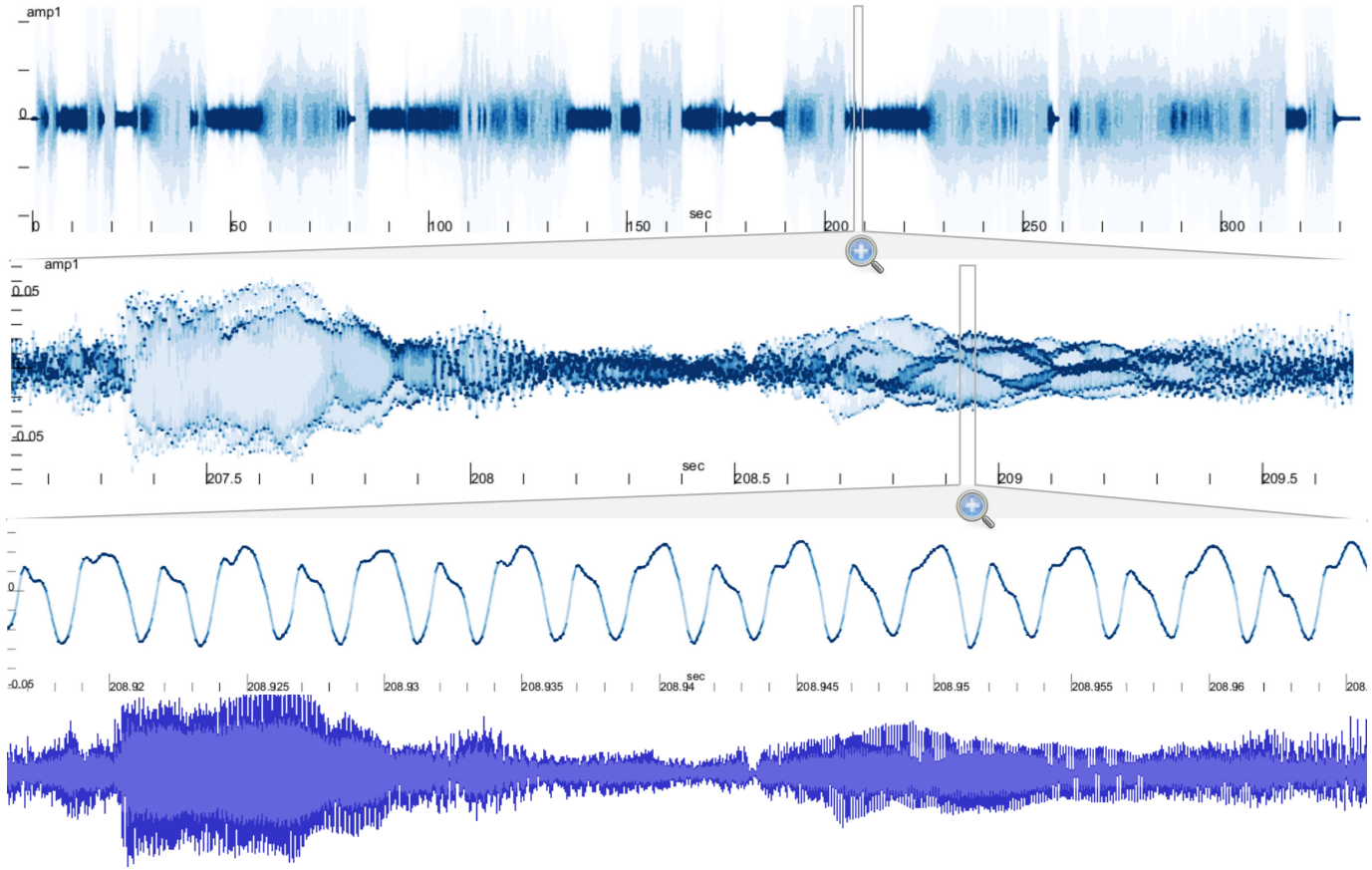

Figure 9: Beethoven, Symphony No. 5 shown using CDE on the full waveform. The top image show five minutes and thirty seconds. The gray box in this top figure shows the timespan zoomed into for the second figure. The second figure shows an interesting feature, spanning three seconds, where a bassoon is playing. The third figure spans 50 milliseconds, zoomed into from the second figure. The bottom figure shows the same span as the second, but using Audacity' viewer [8]

phony No. 5. The top image in this figure displays five minutes and 34 seconds, with almost 15 million samples. We use this example specifically, to show our techniques independence from zoom level. Where the top of Figure 9 showed over five minutes, the next zoom level, in the second graph, spans three seconds. In the third graph, we can directly see the curve, as this graph spans 50 milliseconds. We show this as three discrete zoom levels in this figure, but while interacting with the application, this zooming action is both seamless and smooth. The second and the bottom graph, in Figure 9, both show the same timespan of an interesting piece where a bassoon makes an intricate pattern. This pattern is however, completely lost in the bottom graph, which is the default waveform viewer in Audacity [8]. This intricate pattern is made, in part, by the curve seen in the third zoom level. In this zoom level we see that the curve has two distinct repeating peaks, one with a single mode and the other with two modes, and it is these modes that make up the intricate pattern.

5.2 Prediction Curves

Alan Cox once said:

> I figure lots of predictions is best. People will forget the ones I get wrong and marvel over the rest.

which, somehow, nicely fit the scheme on how modern weather forecasts are done. Instead of a single prediction, an ensemble of possible futures are outlined. However, when a forecast is prepared for public display, it is most often reduced to a single, most likely outcome. When the ensemble of curves spread out, forming a normal distributed pattern, the mode, can correctly be presented as the likely outcome, and the variance, can be presented as the prediction certainty. However, when the ensemble spreads out with two modes, as shown in Figure 5, this model breaks down. We suggest two different use cases for CDE in prediction. In the first case, real-time data is combined with prediction ensemble curves. The historical data will appear as a solid line, and at the most recent sample, an ensemble of curves will possibly spread out, defining the density estimator of the future outcome. This solid line, representing the measured observations is shown in Figure 5, where at $x = 5$ the CDE spreads out into the distribution of future outcomes. By using the full ensemble, rather than the best represented outcome, the operator can also prepare for worst case scenarios, if their probability reaches a given threshold.

The next use case for prediction is repeating, or cyclic patterns. We can find one such cyclic pattern in the yearly temperature. In Figure 10 we show temperature readings, per hour for ten full years by the weather station at Flesland in Bergen, Norway. Data courtesy of eKlima [89]. The temperatures are drawn as ten overlapping curves, using Microsoft Excel in the lower graph. In the middle graph, the moving average over the temperature for all years, and its standard

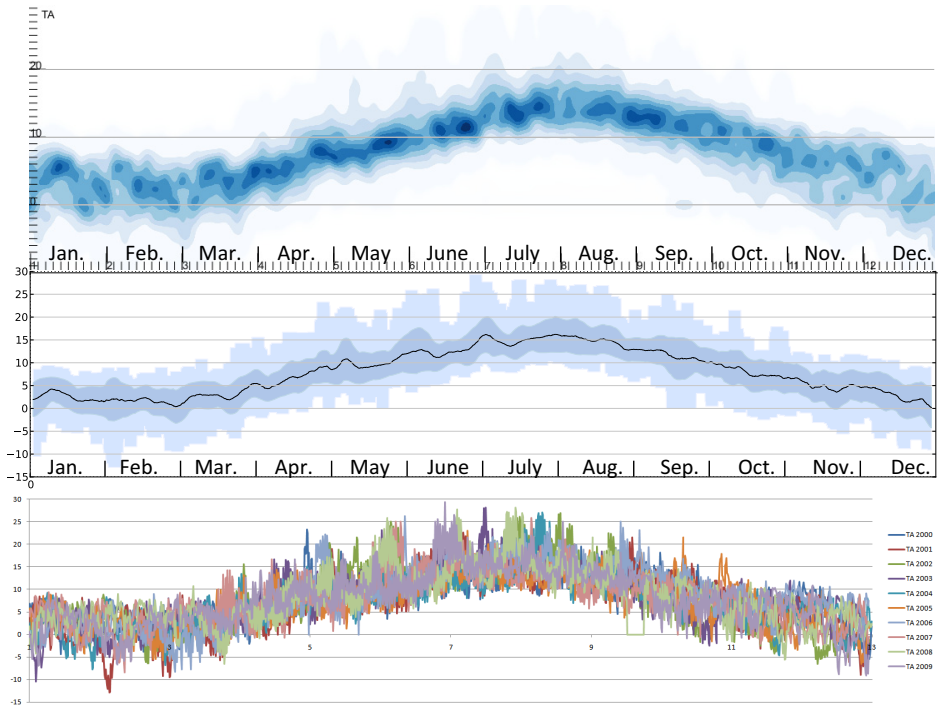

Figure 10: Temperature readings by station nr. 50500 at Flesland, Norway, years 2000 through 2009 with month on the x axis. The top image using CDE, the middle image smoothing using a moving mean and standard deviation, and output by Microsoft Excel™ on the bottom. Notice that the uncertainty/spread of temperature is greater in the winter months Nov. to Feb., than the rest of the year, shown clearly in the CDE. June contains the most stable temperature, represented as the high density there.

deviation is aggregated and shown. The top graph show the CDE for these ten years. By choosing a specific date, the vertical column there represent the probability of which temperatures are likely at that date (according to history). For example, in early June we can see that the probability for the temperature for the EuroVis event is spread out from seven to approximate 18 degrees (given both night and day temperatures). Another way to interpret this CDE, is by looking at the intensity of the highest mode. The higher the mode, the more stable temperature, and vice versa. Historically the temperature is more unstable (less likely to predict a correct outcome) in the winter months, Nov. through Feb. One interesting finding here is the disparity between the mean, as shown in the second graph in Figure 10 and the CDE, for Nov. and Dec., probably due to the less normality of the distribution here.

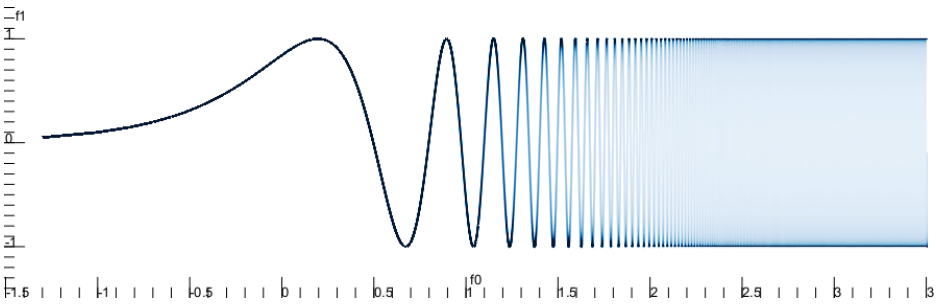

Figure 11: By compressing time with a semi-logarithmic scale, a high level of detail can be read out on recent values, while an overview is available. The logarithmic exponent is given on the x axis.

5.3 Process Visualization

In process visualization, the priorities are often first placed on understanding the *now*, the current situation, followed by both prediction and understanding the historical data. A visualization that receives streaming data, should both emphasize the most recent data, while providing an overview of historical data. Streaming data are, often shown in a temporal window with the most recent values in one end, and the historical data towards the other end. A suitable temporal window is selected, which must be sufficiently small to see the current values, and then data older than this are removed. Figure 11 shows an example use of our CDE with non-linear time, which serves both to emphasize the recent values, to the left, and provide a historical overview that fades into an aggregated probability density estimate.

In drilling, as in many other processes that produce data, we find several data sources that produce bi or multi-modal data. In Figure 12 we show one such example, where the hook-load over time is showed. This image resembles that in our previous work [28], but here it displays time over hook-load, while the other displayed depth over hook-load. Hook load is measured in tonnes, and behaves in a bimodal fashion because the hook is either lifting the entire drill string, or when the drill string is attached in slips to the platform, is zero (actually the weight of the hook itself approx. 40 tonnes). This figure shows the progress over six hours, and we can quickly read out that most of the time has been spent with the drill string attached to the slips, since the mode is highest at 40 tonnes. The second finding, is the spans where the hook load was only at 40, meaning that the operation stalled, and precious time was lost.

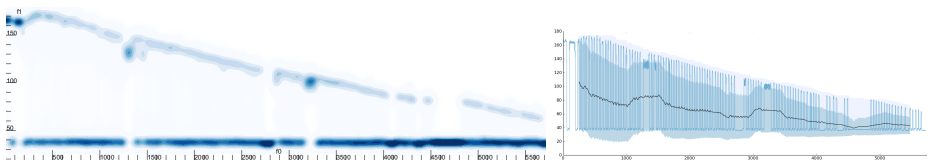

Figure 12: Process data from a drilling operation showing hook-load in tonnes over time in seconds. The right view shows the moving mean, standard deviation and extent, which for this bimodal distribution works particularly bad. The left view displays the curve density estimate of the same data.

6 Summary and Conclusions

We have described the need for a visualization that can represent curves independent of frequencies, zoom level and models, which does not yet exist in the current state of the art. We provide a novel technique on how to render curves independent of sampling-rate, zoom level and curve frequencies. Since kernel density estimates does not impose any model on the distribution, our solution will correctly display data with a single mode, bimodal, tri-modal, or indeed with any underlying model. We have provided implementation details, to promote the usage of this technique in both interactive and real-time settings. Furthermore we have provided several compelling examples of real world usage, showing both where this technique can improve current usages of visualization, but also the versatility of this as a general technique.

For future work, we intend to see how we can use the described technique to provide aid in modeling of data, providing immediate feedback on model suggestions. Furthermore we plan to apply this technique into the daily usage for process visualization, and establish its performance with a user study.

7 Acknowledgements

The work presented here is a part of the project “e-Centre Laboratory for Automated Drilling Processes” (eLAD), participated by International Research Institute of Stavanger, Christian Michelsen Research and Institute for Energy Technology. The eLAD project is funded by grants from the Research Council of Norway (Petromaks Project 176018/S30, 2007-2011), StatoilHydro ASA and ConocoPhillips Norway.

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 INFOVIS*, 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.