A Metric for the Evaluation of Dense Vector Field Visualizations

Victor Matvienko, *Student Member*, *IEEE*, and Jens Krüger, *Member*, *IEEE*

Abstract—In this work, we present an intuitive image-quality metric that is derived from the motivation of DVF visualization. It utilizes the features of the resulting image and effectively measures the similarity between the output of the visualization method and the input flow data. We use the angle between the gradient direction and the original vector field as a measure of such similarity and the gradient magnitude as an importance measure. Our metric enables the automatic evaluation of images for a given vector field and allows the comparison of different methods, parameters sets, and quality improvement strategies for a specific vector field. By integrating the metric into the image-computation process, our approach can be used to generate improved images by choosing the best parameter set. To verify the effectiveness of our method, we conducted an extensive user study that demonstrated the metric's applicability to various situations. For instance, our approach elucidated the robustness of a DVF visualization in the presence of data-altering filters, such as resampling.

Index Terms—Visualization evaluation, texture-based visualization, LIC

1 Introduction

Vector field visualization marks one of the cornerstones of scientific visualization. In numerous examples vector field visualizations have demonstrated their practical use in various scientific and engineering disciplines. Within the realm of available methods, dense vector field (DVF) visualization techniques are known for their ability to give an excellent quick overview of the entire vector field at once. The generation of such a global picture often reduces the risk of missing important features in the flow. Producing a *good* dense visualization, however, does require a careful choice of methods and their parameters. Considering Cabral and Leedom's original LIC work [1], there are already a number of parameters, such as the number of integration steps, the choice of noise, or the filter kernel. Later extensions that dealt with improving the quality added more methods to choose from and parameters to set (see Section 3 for details). In this paper, we present a metric to measure the quality[1] of DVF visualizations. The only input to the computation is the image to be tested and the vector field; no knowledge about the algorithm that is used to compute the image is required. To our best knowledge, this is the first metric tailored specifically to DVFs that compares different approaches not only visually, but can also make an automated quantitative statement. This makes it possible to evaluate a large number of parameters and methods automatically in order to generate better DVF visualizations (see Fig. 1). We also demonstrate that the introduction of a quantitative metric allows for the analysis of more general concepts. For example, we can look at the influence of the random seed on a visualization: even with the same noise model, a different random seed will generate a completely different image for most methods. How much does the seed influence the quality? What happens if a DVF image is downsampled; are flow features still visible? If a visualization is suboptimal, how are the errors distributed in the image? These and other questions can be addressed with our proposed quality metric. The concept presented in this paper is based only on the two local image features: gradient magnitude and the angular error between the vector field and the gradient. We as well give an intuition for interpretation of the metric and for generalization of the presented concepts which might help to construct further task-oriented metrics.

Our algorithm is computationally inexpensive and is well suited for parallel execution on many-core architectures such as GPUs. Even our prototype GPU implementation computes the metric for a 1,024 × 1,024 image in the order of milliseconds on commodity PC hardware.

1. Throughout this text we refer to the quality of a DVF visualization in a general sense meaning a human-perceived characteristic of the image, evaluated by a large group of visualization experts. We are aware that it is not possible to capture all the aspects of the flow visualization in one scalar, but still think that compliance of our metric with experts opinions is somewhat useful for comparing visualization images to one another.

- *V. Matvienko is with Saarland University Cluster of Excellence MMCI, IVDA group, Campus D3.4, Room 2.39, 66123 Saarbrücken, Germany. E-mail: victor.matvienko@dfki.de.*
- *J. Krüger is with IVDA, DFKI, Intel VCI, and SCI, Campus D3.4, Room 2.32, 66123 Saarbrücken, Germany. E-mail: jens.krueger@dfki.de.*

Manuscript received 20 July 2012; revised 12 Apr. 2012; accepted 7 Aug. 2012; published online 15 Aug. 2012.
Recommended for acceptance by G. Scheuermann.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-07-0161.
Digital Object Identifier no. 10.1109/TVCG.2012.170.

2 Our Contribution

In this paper, we propose an intuitive image-quality metric specifically tailored for the evaluation of DVF visualizations. Formally, it is based on the average similarity between the visualization image and the underlying vector field. Local image gradient is taken into account as an indicator (weight) of visual importance. The implicit assumption behind the computation of the average metric

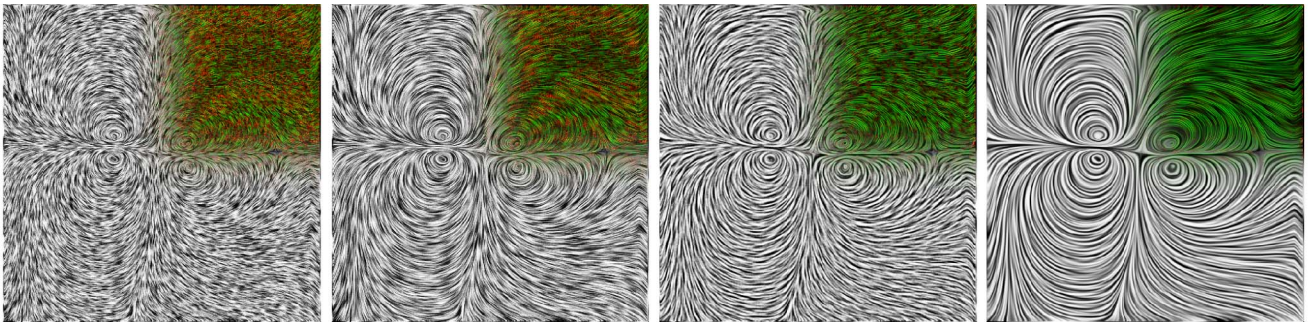

Fig. 1. Four dense vector field visualizations of a vector field, each computed with different parameters. The images are ordered from low to high quality from left to right. The inlay on the top right shows the local metric values, red indicates undesirable image derivatives along the flow on the scale from 0 (black) to 200 (brightest red), green encodes derivatives perpendicular to the flow on the scale from 0 (black) to 100 (brightest red).

value over the whole image is that the visualization quality doesn't vary significantly in different regions, thus making the metric insensitive to such local variations. This fact is illustrated by the example in Section 6.

Since the described measure is mainly meant to quantify the visual/scientific value as perceived by a human, we demonstrate the practical relevance of it with a user study carried out on 53 subjects resulting in about 600 image classifications. The outcome of the survey was a surprisingly high compliance between the metric and the human evaluation.

In this study, we identify two types of users that give answers that contradict the other group, and we propose an adjustment parameter in the metric to make it specific to either group. Finally, with a number of examples, we show how a metric such as ours can be used not only to compare specific images but also to analyze more general phenomena.

The remainder of the paper is structured as follows: In the next section, we recapture the work in flow visualization for the last two decades. Next, we will explain the basic idea of our metric as well as the simplifications that make it useful in practice. In Section 5, we will present a number of experiments carried out with our metric, compare different approaches, and propose new extensions to the LIC method. Next, we will discuss our findings and conclude with a number of ideas for future research.

3 Previous Work

According to Post et al. [2], vector field visualization methods can be classified into four categories: direct, texture-based, geometric, and feature-based methods. Recently, Salzbrunn et al. [3] proposed a fifth category, partition-based techniques. The metric we propose in this work currently specifically targets the texture-based methods; therefore, we will focus this review on that branch of flow visualization and evaluation. For a complete overview of alternative vector field visualization techniques, we refer the reader to the S.T.A.R. by Laramee et al. [4].

Work in the area of texture-based visualization can be categorized into improvements in performance (e.g., many-core/GPU acceleration, methodical enhancements), extensions to novel domains, and quality improvements in these domains (e.g., 3D DVF, flow on surfaces, time dependent/animated visualizations). The roots of texture-based methods date back to the early nineties, with the pioneering work by Van Wijk [5], and Cabral and Leedom [1]. Since then, a vast number of improvements have been proposed, and the first decade of research has also been collected in the S.T.A.R. by Laramee et al. [4] Newer work that focuses specifically on GPU-based methods can also be found in the book by Weiskopf [6]. Even today, there is still room for significant research in this area, and one field of ongoing interest is the concept of double or manifold LIC computations. Introduced by Okada and Lane [7], the concept has been revised and refined by Weiskopf [8] and Hlawatsch et al. [9]. The basic idea is to apply the LIC algorithm multiple times, using the filtered output of the previous LIC pass as input "noise" for the next pass. Generally, this is done to improve performance but it also tends to reduce aliasing problems in the image (we take a closer look at this property in Section 5.5). Any of these approaches can be coupled with advances in other fields, such as the introduction of wavelet noise [10] which provides better input noise fields, resulting in improved LIC images.

As a consequence of the ever-growing number of algorithms in DVF and vector field visualization techniques in general, the demand for the comparison and analysis of existing methods grows. To study the quality of different vector field visualization methods, Laidlaw et al. [11], [12] carried out extensive user studies with experts and nonexperts for 2D vector field visualization techniques. The study was designed to compare fundamentally different visualization techniques (such as dense LIC images and sparse glyph-based techniques). In their studies LIC turned out to be the least effective method. Pineo and Ware [13] took a different approach by using a numerical model of the primary visual cortex of the brain (Visual Area 1) to understand how humans follow the flow, and conducted a user study primarily to verify that simulation. Despite using similar visualization models to Laidlaw et al., they concluded that LIC and equally spaced stream lines are the most effective visualization techniques for this domain. Later, Forsberg et al. [14] conducted a similar study, however, in that study dense methods were not considered.

While all the previously mentioned studies were interested in a comparison of visualizations, they were aimed at providing an accurate answer to the question of which visualization method is better for a given set of tasks. Advances in any of the involved methods would require a new study.

In contrast to these user-centric studies, other researchers have developed models to numerically evaluate the quality of a given visualization. Stalling and Hege [15], as well as Stalling in his thesis [16], did a thorough analysis of the influence of various LIC parameters on signal-processing level. Cui et al. [17] consider the abstraction in multi-resolution information visualization methods, while Chen [18] focused on the abstraction level in network visualizations. Van Wijk [19] proposed an economic benefits and costs model to compute the value of a visualization. In contrast, Filonik and Baur [20] considered the aesthetics of a visualization as a quality metric. Recently, Jänicke and Chen [21] proposed a salience-based metric to evaluate visualizations in general, but also used flow visualizations including a LIC image as a specific example. They evaluate the quality of an image by comparing a salience map with a relevance map that is either user defined or computed from the visualization data with a salience map. Most closely related to the idea presented in this paper is the recently published work by Jänicke et al. [22]. In their work, the idea of using the original vector field to evaluate the visualizations is exploited as well but preceded by visualization-specific reconstruction of the field from the image.

In summary, most of the considered visualization methods improve speed or quality of LIC variations by introducing more functionality to the technique. However, this approach requires a careful selection of the parameter settings. The user surveys—although used extensively to evaluate the general quality of visualization methods—are not applicable for the task of finding a particular optimal parameter values for each and every image. In our work, we want to take a different view on the problem and automatically rate the results of the selection of the parameters and visualization techniques variations.

4 Constructing the Metric

In this section, we present a very general description of our metric and then refine and simplify this formulation to make it useful in practice. As the metric is supposed to express the scientific quality of an image for a human observer, we conducted a user study to train and verify our formulation.

4.1 Basic Idea

The idea of our metric is very straightforward and based on the motivation of LIC images. What the LIC method (as well as other texture-based dense methods) tries to achieve is a low contrast along the path of a integral curve, while at the same time maximizing the contrast perpendicular to that curve.[2] We formulate this property of LIC images as the local inequality (1); desirable to hold in each point of the image

$$\left|\langle\bar{\nabla}f,\bar{v}^{\perp}\rangle\right| > \left|\langle\bar{\nabla}f,\bar{v}\rangle\right|, \tag{1}$$

where $\bar{v}$ is the vector field, f is the intensity of the image, and the term $\langle\bar{\nabla}f,\bar{v}\rangle$ denotes the dot product of the gradient of the image and the vector field. This means that we require

2. Please note that for now we assume that the images are not downsampled for display, and users are capable of viewing the image in its full resolution, i.e., they have sufficient eyesight. We discuss the question of how to deal with downsampled images later in Section 5.

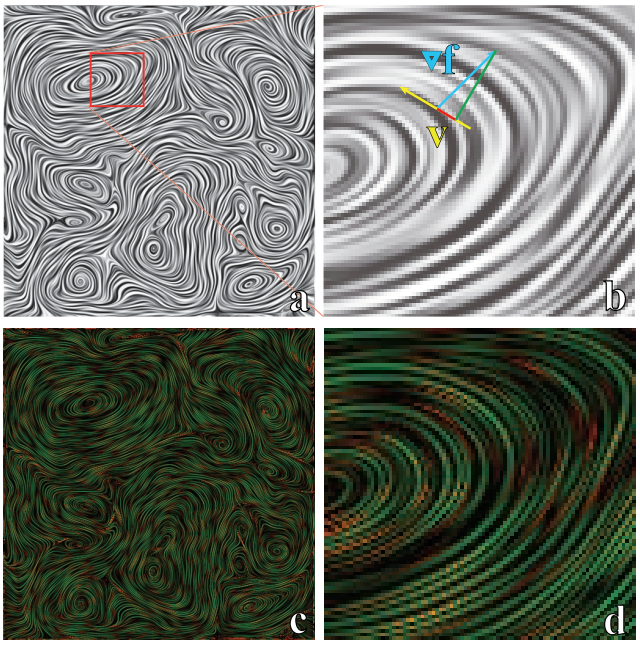

Fig. 2. Image A depicts a DVF visualization with a small section magnified in image C. In this magnified image for a single pixel, the vector field direction (yellow) and the image gradient (blue) are shown. The red and green lines in this image indicate the decomposition of the gradient in the flow direction and perpendicular to the flow. Image D shows these two components for the magnified image and B for the entire domain.

the image-gradient magnitude in the direction of the flow to be lower than the image-gradient perpendicular (see Fig. 2) to the flow. The best visualization under this assumption will be a black and white image depicting continuous lines perfectly aligned with the field and having a width of one pixel and distance of one pixel between them.

Such a visualization is impossible to achieve with existing stream line placement methods, except for the trivial cases (e.g., the constant field), but it would allow users to recognize and follow the stream line contours, which is an important process for understanding the flow, as detailed by Pineo and Ware's work [13].

In our metric $\tilde{M}$, defined by (2) we compute the difference between terms from two sides of the inequality (1) and average over the entire image to get a single value per picture

$$\tilde{M} = \frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} F\left(\left|\left\langle\nabla f,\frac{\bar{v}^{\perp}}{\|\bar{v}^{\perp}\|}\right\rangle\right|\right) - G\left(\left|\left\langle\nabla f,\frac{\bar{v}}{\|\bar{v}\|}\right\rangle\right|\right)d\bar{\omega}, \tag{2}$$

where the point $\bar{\omega}$ is in the image domain Ω where $\|\bar{v}\| \neq 0$, $\bar{v} = \bar{v}(\bar{\omega})$ and $f = f(\bar{\omega})$. Functions F and G are two weighting functions that can be chosen arbitrarily in the mathematical sense, although some choices might have more value for applications than others. Naturally, in our implementation we approximate the integrals with Riemann sums and the gradients are approximated using finite differences (i.e., we and compute a sum over all pixels in the image normalized by the number of pixels). The field is supposed to be given in each point $\bar{\omega}$, but can be of course interpolated if necessary. We normalize the vector field by its norm to

exclude the flow magnitude from the computation. Thus, large positive differences would correspond to the desired high metric value. Negative differences arising in the points where the inequality 1 does not hold would result in a low metric value. Here, we are not trying to interpret the computed quantity $\tilde{M}$ as some inherent attribute of an image. What makes it interesting for the applications is the ordering it produces between several dense visualizations of the same vector field.

4.2 Simplification

The major issue with our metric introduced above is the reasonable choice of two unknown functions F and G. We decided to approach this problem by the principle of *Occam's Razor* [23]. In our case this means to start with the most simple formulation (2) and gradually increase the complexity until the results of the metric match the quality rated by a human observer. Consequently, we started with reducing the functions F and G to a simple weighting factor λ to get the simplified metric M in (3). The parameter λ is supposed to be specific for each user and is to be learned in a survey

$$M = \frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} \lambda\cdot\left|\left\langle \nabla f, \frac{\bar{v}^{\perp}}{\|\bar{v}\|}\right\rangle\right| - (1-\lambda)\cdot\left|\left\langle \nabla f, \frac{\bar{v}}{\|\bar{v}\|}\right\rangle\right| d\bar{\omega}. \quad (3)$$

We expected this simple version of the metric to give somewhat useful results, but as many of the perceptional processes in the human visual system are highly nonlinear, we assumed that we would end up choosing F and G as arbitrary functions that would probably be highly specific to each person. Surprisingly, the results computed from this simple version of the metric resulted in perfectly rated images for our team, with $\lambda = 0.5$ effectively eliminating the weighting term and thus making the formulation even simpler. This means, that given sets of images that were computed by randomly varying a number of parameters (e.g., integration length, choice of noise, histogram equalization) the ordering by metric values matched the ordering that our team had chosen.

4.3 A Different View on the Metric

In the inequality (1) and the derived metric (3), we considered two local quantities of the DVF image combined: the direction of the vector field and the gradient of the image.

Now, let us formulate a metric, using the image gradient magnitude and the angle between the gradient and the vector field.

Since we assume that the direction perceived from the image is orthogonal to the image gradient it makes sense to measure the angular error between these two directions. The image gradient magnitude, can be thought of as the confidence of the direction. If the gradient is low we will hardly be able to perceive the direction in the image but if it is high we would want it to indicate the right direction. This observation allows us to use the gradient magnitude in a weighting factor for the angular error, resulting in a following generic formulation of the metric

$$M = -\frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} W(\|\bar{\nabla} f\|)\cdot K(\alpha) d\bar{\omega}, \quad (4)$$

where $\alpha = \alpha(\bar{\omega}) = \angle(\bar{\nabla} f(\bar{\omega})^{\perp}, \bar{v}(\bar{\omega}))$, K is the generic error factor, measuring the angular error and W is the generic confidence factor, depending on the image gradient. The minus sign in front of the sum denotes that the quality of the image is higher for the lower error.

The function W and K can be chosen with respect to the specific task.

If one is interested in the overall contrast level which is clearly dependent on the gradient magnitude the simplest form for the W factor is $W(x) = x$. It also might be useful to normalize the weights over the whole image domain if one is not interested in distinguishing low- and high-contrast images. For the angular error factor K the simplest natural choice is $K(x) = |x|$. With these two definitions for W and K the metric (4) becomes a weighted L_1 norm of the angular error in

$$M = -\frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} \|\bar{\nabla} f\|\cdot|\alpha| d\bar{\omega}. \quad (5)$$

As our user study in Section 4.4 shows there might not be a unique metric suitable for all users. Consequently a parameter is required to adjust the metric to user preferences. For the metric formulation (4) the natural choice would be to use a function space of L_p norms with parameter p as in

$$M = -\frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} \|\bar{\nabla} f\|\cdot|\alpha|^p d\bar{\omega}. \quad (6)$$

Now, let us take a second look at the original simplified version of the metric with parameter λ in (3) and bring it to the generic form (4). With a few simple steps, we can separate the gradient magnitude and the angular error. To achieve that we rewrite the dot products between the gradient vector and the vector field direction using cosines of the angle between the vector field the gradient, obtaining the

$$M = \frac{1}{|\Omega|}\int_{\bar{\omega}\in\Omega} (\lambda\cdot|\cos(\alpha)| - (1-\lambda)\cdot|\sin(\alpha)|)\cdot\|\bar{\nabla} f\|\, d\bar{\omega} \quad \text{where } \alpha = \angle\big(\bar{\nabla} f^{\perp}, v\big). \quad (7)$$

In the Fig. 3, we illustrate the shape of some frequently used L_p norms as well as the angle factor in the original formulation (7) with different λ. The important observation is that varying λ produces similar curves as commonly used norms, this similarity is also reflected in the quality of the metric, i.e., how well it can predict the users choices after the learning phase (in this case tweaking the exponent). Our experiments indicate no significant improvements of the prediction quality when using (6) as metric over our initial choice (7).

Since we have no evidence that one particular type of K is better than the other we suggest to take the computational efficiency into account. For performance reasons it is generally a good idea to avoid the computation of a integral over the whole image to obtain the metric value for every new K. The original formulation (3) allows the precomputation of the sums over the image separately. This makes the

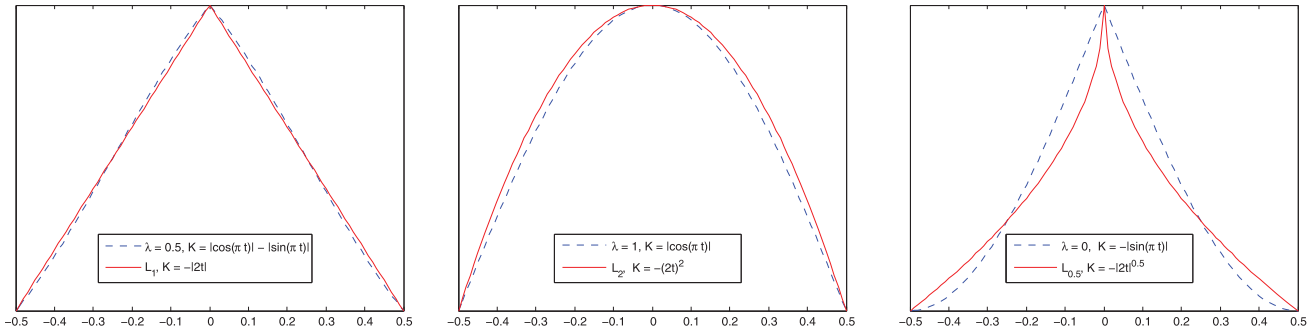

Fig. 3. The shape of K factor in (4) for functions producing common norms and for different values of λ in (7). Note that the functions are biased and rescaled appropriately for display in this image.

metric M a linear function of the parameter λ, that can be easily computed for different λ.

4.4 Training and Verification of the Metric

As the experiments of the informal evaluation within our group of five (probably biased) researchers were not sufficient to consider the metric applicable for a larger population, we decided to extend the experiment to a much larger group of independent subjects. The idea of the survey was to show the relevance of the computed quantity to subjective evaluation criteria used by visualization experts and to adjust the metric parameter for better performance. For this purpose, we designed a browser-based survey platform.

4.4.1 Survey Setup

To conduct the survey, we selected over 2,300 LIC visualizations of 11 flow fields from our database containing over half a million images. Most of those flow fields were also used for the illustrations in this paper. Those included LIC images with varying kernel length (1 to 200 steps), kernel shapes (box, Gaussian), postprocessing filters(high pass, histogram equalization) on each of the manifold LIC iterations as well as the different number of iterations (1 to 3). The images were selected to cover a wide range of methods and settings. The survey database was then accessed by a web service we developed for this paper. To acquire participants, we sent invitation e-mails to a number of experts in the field of flow visualization. The users were not provided with any compensation for performing the study.

We designed the survey webpage as follows (see Fig. 4): After a brief introduction to the survey itself, each of the subjects was asked once if they considered themselves experts in the field of flow visualization and wanted to participate in the scientific visualization study, or if they preferred to judge the images purely by their artistic merit. We added this question because of the anonymous nature of our survey: we could not prevent people from forwarding our invitations to others. For this reason, we decided to encourage the recipients of our mails to forward the survey to their friends and colleagues.

In the actual survey, the subjects were presented with a set of four randomly chosen DVF visualizations of the same flow field. The task was to rate the images from 1 to 10 on either scientific or artistic merit. The subjects could evaluate as many sets of images as they liked. We deliberately didn't pose any domain-specific tasks to be accomplished by the participants with the help of the images, since we believe that the general usefulness of the visualization as estimated by the experts is a more beneficial yet harder to formalize property of the visualization than its suitability for solving some subjectively chosen problem. Our hypothesis was that our metric, perhaps with some minor adjustment of parameter λ, would produce the ordering of images close to the one provided by the subjects based on their expertise in the domain of scientific visualization.

The survey is still available and can be accessed at http://survey.ivda.uni-saarland.de.

4.4.2 Survey Results

By the time we did the evaluation, we received a total of 596 votes from 53 distinct users. To ensure that those are likely to be distinct persons we used both the IP address and a tracking cookie, which we set on the user's machine. From these users, 37 designated themselves as experts in the field, and 16 chose to evaluate the images by their artistic merit.

From these answers, we derived from each set of four images $\binom{4}{2} = 6$ orderings, to get a total of 3,576 orderings. Along with these, we also stored the differences in rating as a measurement of the user's confidence of this ordering.

To perform the training and verification of the metric, we use the stratified holdout method with a $\frac{2}{3}$ to $\frac{1}{3}$ training to test-set split. From the training set, we learned a $\lambda = 0.26$ for the metric, which resulted in 91 percent matches in the test.

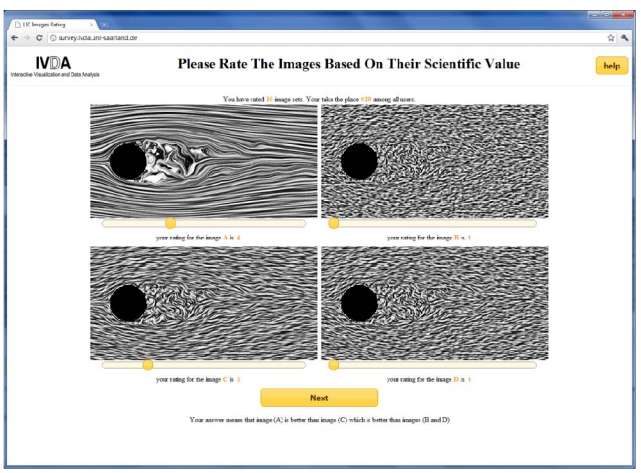

Fig. 4. A screen-shot of the web survey. The user is asked to rate four images from 1 to 10.

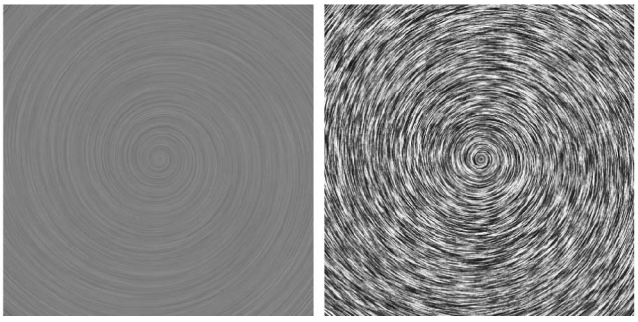

Fig. 5. An example of the ambiguous low versus high-contrast case. If you prefer the left image over the right, then you should use a $\lambda = 0.1$ otherwise a $\lambda = 0.4$.

Since we have a relatively large test set we can assume that this score generalizes well. With a confidence level of 95 percent we got a confidence interval of [0.88; 0.93] applying the scheme as proposed by Kohavi [24].

In the next step, we looked at all the images pairs where the metric ordering disagreed with the user's choice, and we found that those image pairs fell into two categories:

1. Images with metric values different by 5 percent or less. Those were mostly low-quality images, hardly recognizable as flow visualizations at all. In this case, the confidence value from the user was also low (rating difference of only one) and we assume that the participants in most cases just saw these images as *bad images* and the classification was relatively arbitrary.
2. In disagreement with our metric's evaluation, low-contrast images were sometimes rated better than contrast-enhanced images with less coherence along the flow by some participants (see Fig. 5). Upon further investigation, we discovered that for these image pairs, about half of the participants' votes preferred the low-contrast images and one half that prefers high-contrast images. Consequently, it was impossible to adjust the parameter λ to get a perfect match for all votes. We concluded that there are two types of people, one half that prefers the low-contrast images and the others. Therefore, we decided to split the votes into those two types. As a regularization, we added the voters that never encountered a set with low-contrast images to both groups. Within each group, we used the same training/test splits as detailed above. Then, we optimized the parameter λ independently for both groups, resulting in a $\lambda = 0.1$ for the "low-contrast group," which now matched 93 percent of those votes in the "low contrast and regularization group." For the "high contrast" group, a $\lambda = 0.4$ setting resulted in 98.9 percent. To solve this ambiguity in practice, we propose a small training set of one or two images pairs (e.g., Fig. 5) to determine in which category a user falls. An application could then adjust λ accordingly.

To our best knowledge, the second observation we made has not been described in the literature before. While we do have a few hypotheses about possible reasons why some participants preferred one type of image or the other, we cannot offer a distinct explanation. As our survey was carried out anonymously, we cannot conclude whether this difference results from cultural background or past experience, or if it is rooted in differences in eyesight, psychological reasons, or any of a multitude of causes. As we think that this question might be interesting for future visualizations, we are planning to design an additional study devoted specifically to this issue (see Future Work section).

5 Metric Application Examples

The metric, as trained by the user study, can now be used to compare two images of the same data set computed with different DVF methods or parameter sets. Our experiments have shown that absolute values of the metric can also be used to make a general statement about the image quality. Throughout the paper the image values are supposed to be normalized between 0 and 1 and the metric value is multiplied by 200. The scaling factor of 200 was chosen to get "nicer" values and is motivated by the observation that the best images we were able to produce often range around an unscaled metric value of roughly 0.5; therefore, we upscaled the values to get numbers that we think of as a percentage, though values larger then 100 percent are possible. The images with a metric above 70 were generally considered high quality. We used this scale for all the graphs in the paper. For the computation of all the graphs in this section we set λ to 0.26 but note that although the shape of the graphs changes, the conclusions we draw are the same for the specific low- and high-contrast groups.

In the remainder of this section, we present a number of scenarios where we use the metric not only to compare two images, but also to look at more general issues. Some of our results are well known and are just presented to demonstrate the applicability of our method. All of these experiments share the same idea that an automated mechanism "looks" at a large set of DVF images and extracts images with interesting statistical properties (such as the best or worst) or considers the statistics itself.

5.1 The Per Pixel Metric Distribution

In this experiment, we are going to take a closer look at the distribution of the local metric values. In the other scenarios, except for the manifold LIC, we only considered the average over the entire image to compare different techniques. To further analyze a specific technique for a given data set, however, it may be interesting to visualize where it achieves a high score and where it fails. This could be used as a local confidence in the flow visualization or as a clue for how to improve a given visualization algorithm. Various figures in this paper show a number of DVF visualizations side by side to the corresponding metric images. In these metavisualizations in Figs. 6a, 6b, 6c, and 6d it can be seen how the metric captures problems in the LIC image. Visualizations in Figs. 6a and 6b suffer from aliasing, appearing as red dot pattern in the metric image. This problem can be avoided with a twofold LIC or a usage of band-limited noise (e.g., wavelet noise (Figs. 6c and 6d)). The locality of the metric is an especially valuable feature in this case, since the aliasing doesn't have a great influence on the integral metric value if the percentage of aliased pixels in the image is small. On the other hand those pixels can be easily

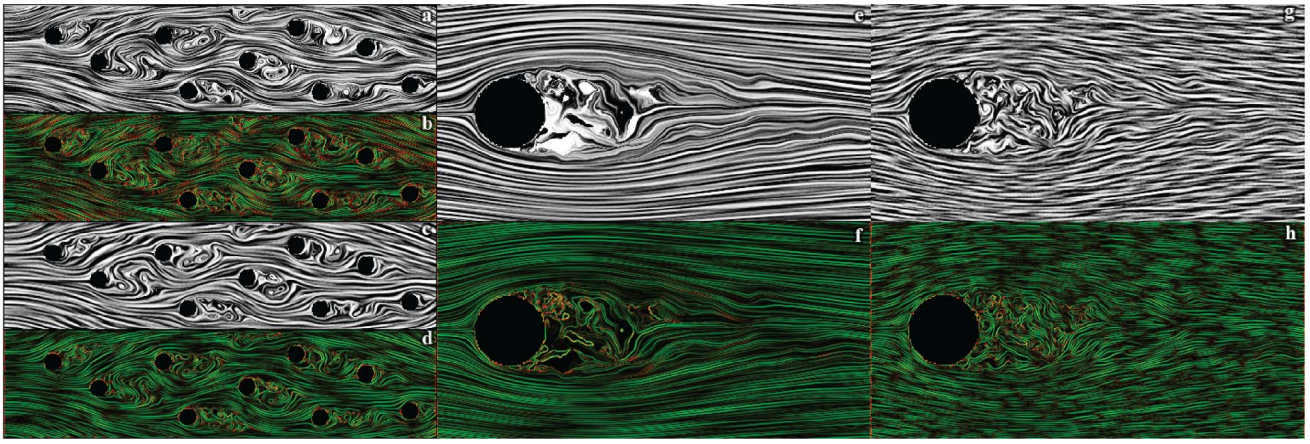

Fig. 6. The images a-d on the left demonstrate how the metric nicely captures aliasing artifacts (in red). The four images e-h, in the center and on the right, show that a constant step-count LIC-method either captures laminar regions accurately (with a large kernel) or turbulent regions (with a smaller kernel). The successive letters of images are paired (DFV image and its metric image).

recognized, since they result in regions in the metric image having extremely low value.

In Figs. 6e, 6f, 6g, and 6h, the influence of the number of integration steps on laminar and turbulent regions can be seen. In Figs. 6e and 6f, 200 steps were used, compared to 60 in Figs. 6g and 6h. We observe that longer traces improve the metric in laminar regions, while shorter traces improve the quality in turbulent regions. This observation can be used to implement an optimized LIC algorithm, as described below.

5.2 Influence of the Random Seed

Most of the DVF visualization techniques rely on a noise field somewhere in the algorithm. While the types of noise vary from method to method, one input to the algorithms is a (pseudo) random number generator. Naturally, these generators produce different outputs depending on the random seed, which may be controllable by the algorithm or not. Either way, a different random seed will produce a different image and at a very fine scale this variation will influence the quality of the image. Therefore, we consider this random seed influence a background noise in the visualization, and we decided to quantify this impact on the image quality. To test this, we conducted the following experiment: we generated 100 distinct white-noise textures, and for each of them, computed a number of line integral convolution visualizations of varying kernel length l from 0 to 200.

For each l we obtained a set of 100 integral metric values of the visualization of the same flow, depending on the noise textures (different for 100 different random seeds). From these 200 distributions of 100 metric values each, we computed the mean and variance of the metric before and after histogram equalization. The results for one flow field are shown in Fig. 7. Thus, each plot aggregates the statistics for 20,000 images. We also studied the behavior of the LIC images without equalization, as shown in Fig. 7a. We can see that starting from noise (LIC with zero steps) with increasing step count we observe a rapid improvement in metric values followed by a quick decline. This happens because the images converge to uniform gray. In the second graph in Fig. 7b, we can see the influence of the random seed in the results. We show this by the means of the variance and the difference between the lowest and highest values of the metric with varying random seed for different kernel sizes. It is relatively small, even for small kernels, and decreases even more with increasing kernel length. This can be explained as the image converges to the mean noise value, which is a property of the noise itself that should be independent of the random seed.

In contrast to these results, the histogram-equalized LIC behaves differently. First, we observe that for this field the mean metric value does not have a maximum inside the tested range but, seems to converge to a relatively high metric value (see Fig. 7c). Continuing the tests with even larger kernels shows that for very large kernels this curve will eventually drop too, though much later. The more

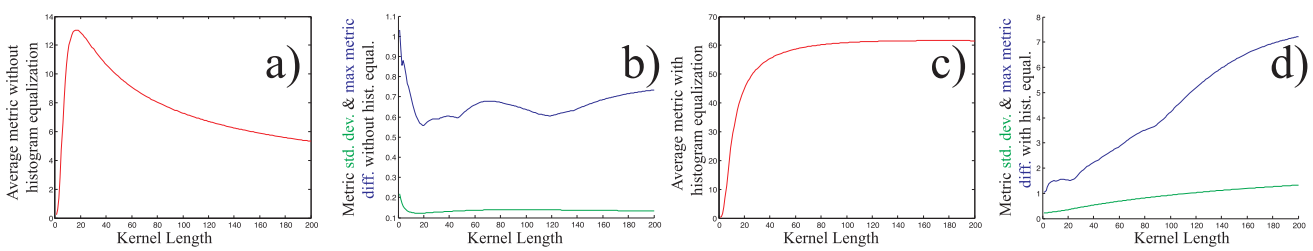

Fig. 7. Graph (a) shows the behavior of the average metric value for the visualizations of the vector field shown in Fig. 8 when only LIC without histogram equalization afterwards is performed. In graph (b) the maximum quality difference in metric, as well as the standard deviation between different random for seeds in relation to the LIC kernel size, can be observed. The graphs on the right show the same information for histogram-equalized images.

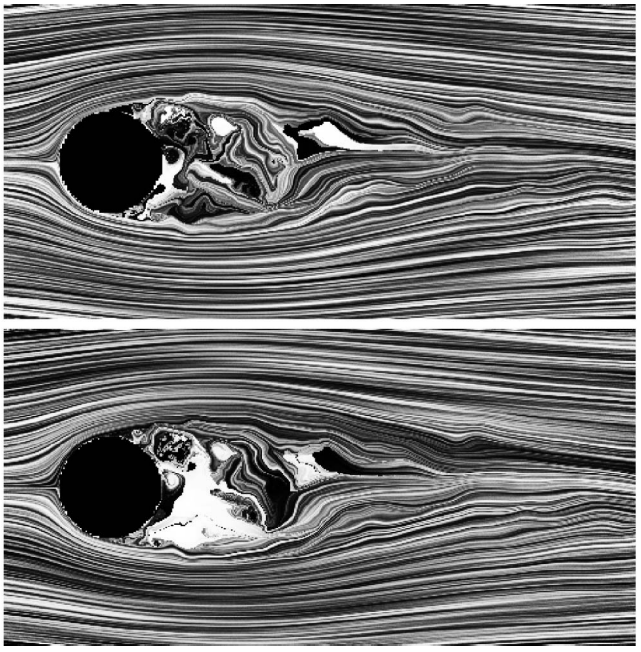

Fig. 8. Two images with the same LIC parameters but different random seeds. Notice the quality differences especially in the turbulent region.

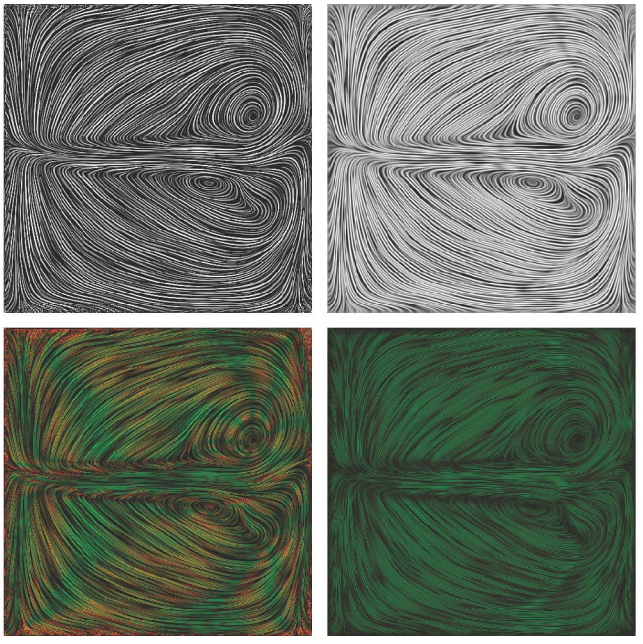

Fig. 9. The image on the left was computed with the *llhe* sequence proposed by Okada and Lane [7]. The right image was computed using the sequence *lelhele*.

important observation, however, is that the metric variance is about an order of magnitude larger then before, and it increases (seemingly linearly) with the kernel size (see Fig. 7d). This can be explained by the fact that the histogram-equalization step, with increasing LIC kernel length, has to stretch smaller and smaller variations onto the full range of grayscales. Consequently, we observe a growing influence of the random seed. This can lead to a difference between the best and worst image of up to 15 percent, which results in visible quality differences (shown in Fig. 8).

Overall, we conclude that in most situations the random seed has little influence on the image quality, as expected, but if intermediate steps of the computation process are significantly upscaled by operations such as histogram equalization, then random seed can have a significant influence on the image quality. It is worth noting that while the mean-value graphs behaved differently for other vector fields, we observe the same variance and maximum variation characteristics for all tested vector fields.

5.3 Manifold LIC

We then decided to take a closer look at the LIC enhancements proposed by Okada and Lane [7]. While discussing a number of other LIC improvements, they proposed using a combination of LIC (l), high-pass filtering (h), and histogram equalization (e) to improve LIC image generation speed and image quality. They further suggest that the sequence *llhe* (meaning: first start with a LIC image, then use that image as noise input to a second LIC computation, next perform high-pass filtering, and finally do a histogram equalization) produces the best results. Next, they mention that they "found no appreciable difference in executing the LIC algorithm more than twice." With our metric, we automate the process of evaluating a large number of LIC and filter parameters and combination of these settings for double, triple, or even higher iterations.

With an extensive parameter exploration approach evaluating tens of thousands of combinations we can confirm that the *llhe* in fact results in the best metric value for our data sets when only two LIC passes are performed. In addition, we discovered that it is possible to improve the quality by applying more operations. We tested combinations of up to four LIC applications with and without interleaved high-pass filtering and/or histogram equalization. In this test we, found that, for example, with a triple LIC sequence *lelhele* we can generate visualizations with an about 15 percent higher metric value (see Fig. 9).

5.4 An (Adaptive) Metric as Part of a DVF-Visualization Pipeline

An obvious use of the metric apart, from simply comparing two images, is to integrate it into a visualization pipeline, instead of choosing a fixed method and parameter set for all images, as it is usually done. An integrated metric can optimize these parameters for each image. Therefore, the visualization system would need to support a number of different DVF techniques and using parameter exploration would search for the optimal image. Intermediate best results could be present to the user as the system optimizes.

As an extension to this fully automated system we, envision a semiautomatic steering approach to further improve the image quality. In this extended version the user is given the ability to use a brush-like metaphor to indicate interesting regions in the intermediate result. Within the metric computation we can use this *focus image* as a scale for the metric. As a consequence, the metric will automatically steer the computation of the DVF visualization toward parameters that produce better results in those specific regions. While the basic concept of a user-defined or vector-field based scalar parameter has been proposed

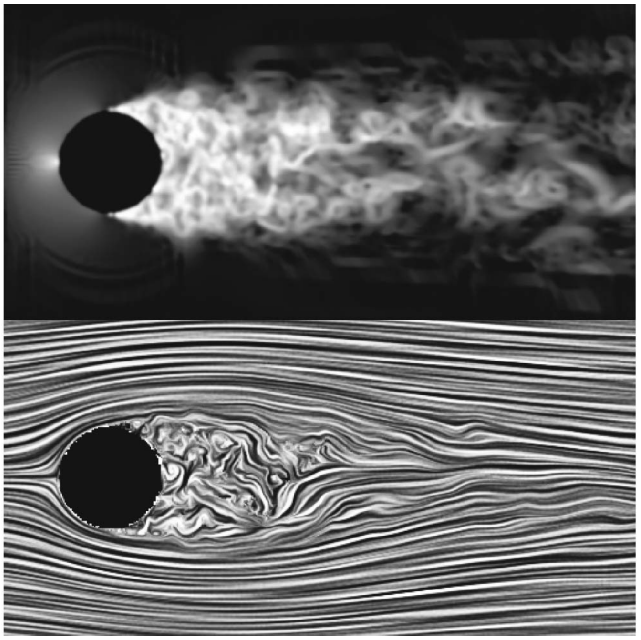

Fig. 10. The top image represents the variance of the flow direction. In the bottom image, we use this weight texture to scale down the step-count and achieve a better overall visualization. Compare this image to the results in Fig. 6.

before [21], [25], [26], [27], the connection with an automated metric and a progressive interactive optimization is a novel concept to our best knowledge.

With this method, we generated the images in Fig. 6 and there we highlighted the laminar regions first, then the turbulent regions.

If this focus and context visualization is not desirable, the findings of the optimization process can also be used to improve the DVF itself. From Fig. 6, we conclude that LIC produces better images if shorter kernels are used in turbulent regions and longer kernels are used in laminar parts of the domain. To verify this, we added functionality to our LIC implementation to compute the gradient magnitude of the of the flow direction (angle) as an approximation of the turbulence, and used this value to scale the LIC kernel (normalizing the obtained quantity between zero and 1). This results in a unique method that achieves a better image than with a uniform kernel (see Fig. 10).

5.5 Image Robustness

In our final experiment, we look at the changes in the metric if the DVF-visualization image undergoes image transformations. In this example, we do not transform the domain and then re-run the visualization algorithm, but we compute the image once and then see what happens when it is transformed. Here, we specifically focus on image rescaling, but other operations such as quantization fall into the same category. For practical applications, it may be interesting to generate an image that is robust to such transformations, specifically if you have control over them. An example is the generation of a LIC image for usage on a webpage or a newspaper. When asked to generate such a visualization, it may not be clear at what resolution this image will be integrated into the webpage or at what resolution/quantization it is printed. As an example, we looked at different LIC algorithms and computed how their metric values behave under subsampling. For instance, as mentioned in [7], twofold LIC produces smooth solid stream lines representing the flow field on a coarse scale. The resulting visualization (Fig. 11 bottom row) is thus robust against downsampling. While ordinary LIC images are much more sensitive to minification (Fig. 11 top row), both these observations agree with the behavior of the metric.

6 Discussion

In this paper, we proposed a novel metric to evaluate the quality of DVF visualizations. We conducted a web-based user study to verify the applicability of our metric and to fine-tune the metric parameter. In the course of this optimization, we discovered two distinct groups of users. To be able to correctly rate the images for both groups, we proposed two parameter settings and a simple—one image pair—test to determine which group a person belongs to. With this distinction, our metric agrees with up to 99 percent of all user evaluations, and we felt comfortable enough to use it in an automated process and draw conclusions in a number of experiments in which we proposed to compare existing methods, validate previous findings, and optimize visualizations.

For the survey and experiments, we implemented a number of different DVF techniques, and in the course of those implementations, we naturally spent a significant amount of time debugging. Thereby, we discovered that

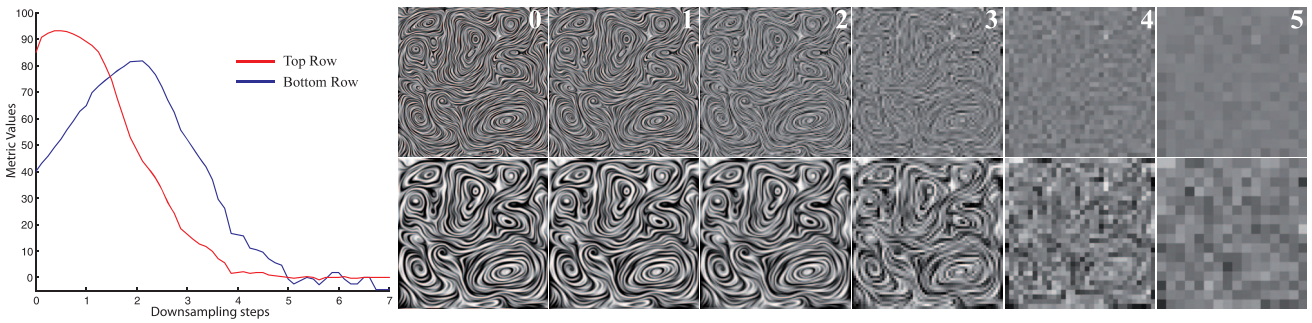

Fig. 11. In the top row, a fine detail LIC image is down-sampled. The resulting sequence suffers from aliasing and a loss of contrast, consequently the metric value (red) degrades quickly. In the bottom row, a LIC image with coarser structures was generated. Its metric value (blue) first increases until it also degrades, but still achieves better quality than the first image. The reason for this metric this behavior is that the gradient is computed using finite differences on the image grid, and thus increases under downsampling for low frequent images, while degrading for high-frequent images. In our opining this corresponds to subjective quality impression of the visualization.

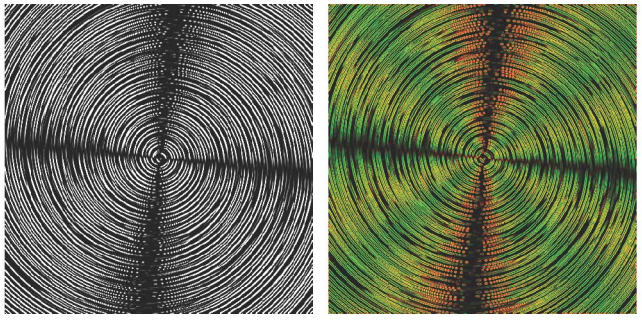

Fig. 12. The left image shows a visualization of a spiral flow generated with the manifold LIC method and uses a poorly implemented high-pass filter in the process. As can be seen on the right, while the metric picks up some of the distortions, the zero-metric areas in black do not significantly contribute to the average value to degrade this image's value.

our metric is also an excellent verification/debugging tool as in most cases when we had issues with a new method this resulted in significantly lower metric values then for previous methods. So far, we encountered only one exception, which was an issue related to a high-pass filter that we applied during a multipass LIC computation; this example demonstrates a limitation of our method. For the visualization shown in Fig. 12, we computed a high metric value. Further investigation revealed that the problem in this image was that the majority of pixels achieved high metric values and black regions do not significantly reduce the average. In fact, if these regions would be distributed differently, we would probably rightly consider this image of good quality. The problem is a global phenomenon—namely, that the black cells cluster in an arbitrary pattern and cause disturbing structures. To tackle this issue, we are considering a hierarchical application of our metric, similar to the idea of Jänicke and Chen [21] for the salience-based metric.

7 Future Work

While working on our DVF metric, we discovered a number of new research questions, some of which are directly related to our metric others are ideas discovered while using the metric.

There are natural extensions in dimensions to consider, along with the applicability to animated methods, time-dependent flow, 3D flow, and flow on surfaces. For time-dependent data, we are considering to add an additional regularization term to the metric to capture requirements such as continuity in the images. For 3D vector fields, we can certainly compute the metric for the volume, but the volume-rendering process also needs to be considered. One possible solution may be the application of the metric to the current view, but a number of open questions will have to addressed here. For flow on surfaces, we expect that the straightforward application of our metric to a given view should already work nicely, but we have not yet investigated this scenario. We also intend to consider sparse vector field-visualization methods. A helpful method for this scenario, as well as for dealing with the failure case mentioned above, could be the hierarchical computation of the metric as mentioned in Section 6.

Another research problem, which will give more value to the proposed metric, is the normalization of the obtained value. For now we are able to compare the values for different images, but avoid interpreting them as a characteristic of an image meaningful on its own. Yet, we realize that a metric comparable for different flow fields and visualization techniques might be useful as a universal scale of the image quality.

There are also new research directions that we discovered during the study, which are not necessarily related to the metric but might be resolved within its parameters. One of those cases is the question of why we observe two classes of users. Here, we are planning a nonanonymous survey in an attempt to find out the reason for this fragmentation. We are also planning to release the database of our current survey after anonymization (replacing IP addresses and cookie by an ID) to the public for further analysis.

Finally, we think that the web-based survey system developed for this paper—while not a completely novel idea—can be useful for other projects as well. With an easy-to-use method, we were able to reach a relatively large number of participants. Therefore, we have begun to integrate a number of additional features such as image zoom, extensions to other media such as videos, an extensive administration back end to supervise the survey and get statistics in real time. As all of the involved components are open-source frameworks, we are going to release the entire system as open source. Additionally, we are also considering to offer our servers as a service for others, which may be helpful for submissions that require a double-blind process that prohibits the use of the author's servers.

Acknowledgments

The work presented in this paper was made possible in part by the NIH/NCRR Center for Integrative Biomedical Computing, P41-RR12553-10 and by Award Number R01EB007688 from the National Institute of Biomedical Imaging and Engineering, as well as the Intel Visual Computing Institute and the Cluster of Excellence "Multimodal Computing and Interaction" at the Saarland University. The content is under sole responsibility of the authors. The flow around a cylinder is courtesy of Octavian Frederich [28].

References

[1] B. Cabral and L.C. Leedom, "Imaging Vector Fields Using Line Integral Convolution," *SIGGRAPH '93: Proc. 20th Ann. Conf. Computer Graphics and Interactive Techniques*, pp. 263-270, 1993.

[2] F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramee, and H. Doleisch, "The State of the Art in Flow Visualisation: Feature Extraction and Tracking," *Computer Graphics Forum*, vol. 22, no. 4, pp. 775-792, 2003.

[3] T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann, "The State of the Art in Flow Visualization: Partition-Based Techniques," *Proc. Simulation and Visualization (SimVis)*, pp. 75-92, 2008.

[4] R.S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F.H. Post, and D. Weiskopf, "The State of the Art in Flow Visualization: Dense and Texture-Based Techniques," *Computer Graphics Forum*, vol. 23, pp. 203-221, 2004.

[5] J.J. van Wijk, "Spot Noise Texture Synthesis for Data Visualization," *SIGGRAPH Computational Graphics*, vol. 25, pp. 309-318, July 1991.

[6] D. Weiskopf, *GPU-Based Interactive Visualization Techniques*, Mathematics and Visualization. Springer-Verlag, 2006.
[7] A. Okada and D. Lane, "Enhanced Line Integral Convolution with Flow Feature Detection," *Proc. SPIE Visual Data Exploration and Analysis IV*, vol. 3017, pp. 206-217, 1997.
[8] D. Weiskopf, "Iterative Twofold Line Integral Convolution for Texture-Based Vector Field Visualization," *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, series Mathematics and Visualization, pp. 191-211, Springer, 2009.
[9] M. Hlawatsch, F. Sadlo, and D. Weiskopf, "Hierarchical Line Integration," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1148-1163, Aug. 2011.
[10] R.L. Cook and T. DeRose, "Wavelet Noise," *Proc. ACM SIGGRAPH Papers*, pp. 803-811, 2005.
[11] D.H. Laidlaw, J.S. Davidson, T.S. Miller, M. da Silva, R.M. Kirby, W.H. Warren, and M. Tarr, "Quantitative Comparative Evaluation of 2D Vector Field Visualization Methods," *Proc. Conf. Visualization '01*, pp. 143-150, 2001.
[12] D.H. Laidlaw, R.M. Kirby, C.D. Jackson, J.S. Davidson, T.S. Miller, M. da Silva, W.H. Warren, and M.J. Tarr, "Comparing 2D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 1, pp. 59-70, Jan. 2005.
[13] D. Pineo and C. Ware, "Neural Modeling of Flow Rendering Effectiveness," *Proc. Fifth Symp. Applied Perception in Graphics and Visualization (APGV '08)*, pp. 171-178, 2008.
[14] A. Forsberg, J. Chen, and D. Laidlaw, "Comparing 3D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1219-1226, Nov./Dec. 2009.
[15] D. Stalling, "Fast Texture-Based Algorithms for Vector Field Visualization," PhD dissertation, Zuse Inst. of Berlin, 1998.
[16] D. Stalling and H.-C. Hege, "Fast and Resolution Independent Line Integral Convolution," *SIGGRAPH '95: Proc. 22nd Ann. Conf. Computer Graphics and Interactive Techniques*, pp. 249-256, 1995.
[17] Q. Cui, M. Ward, E. Rundensteiner, and J. Yang, "Measuring Data Abstraction Quality in Multiresolution Visualizations," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, pp. 709-716, Sept./Oct. 2006.
[18] C. Chen, "Measuring the Quality of Network Visualization," *Proc. ACM/IEEE-CS Fifth Joint Conf. Digital Libraries (JCDL '05)*, p. 405, 2005.
[19] J.J. van Wijk, "The Value of Visualization," *Proc. IEEE Visualization (VIS '05)*, pp. 79-86, 2005.
[20] D. Filonik and D. Baur, "Measuring Aesthetics for Information Visualization," *Proc. Int'l Conf. Information Visualisation*, pp. 579-584, 2009.
[21] H. Jänicke and M. Chen, "A Salience-Based Quality Metric for Visualization," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1183-1192, 2010.
[22] H. Jänicke, T. Weidner, D. Chung, R.S. Laramee, P. Townsend, and M. Chen, "Visual Reconstructability as a Quality Metric for Flow Visualization," *Computer Graphics Forum*, vol. 30, no. 3, pp. 781-790, 2011.
[23] Wikipedia, "Occam's Razor," http://en.wikipedia.org/wiki/Occam's_razor, 2011.
[24] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proc. 14th Int'l Joint Conf. Artificial Intelligence*, pp. 1137-1143, 1995.
[25] M.-H. Kiu and D.C. Banks, "Multi-Frequency Noise for LIC," *Proc. Seventh Conf. Visualization '96*, pp. 121-126, 1996.
[26] A. Sanna, C. Zunino, B. Montrucchio, and P. Montuschi, "Adding a Scalar Value to Texture-Based Vector Field Representations by Local Contrast Analysis," *Proc. Symp. Data Visualisation (VISSYM '02)*, pp. 35-41, 2002.
[27] D. Weiskopf, G. Erlebacher, and T. Ertl, "A Texture-Based Framework for Spacetime-Coherent Visualization of Time-Dependent Vector Fields," *Proc. IEEE 14th Visualization (VIS '03)*, pp. 15-, 2003.
[28] O. Frederich, E. Wassen, and F. Thiele, "Prediction of the Flow Around a Short Wall-Mounted Cylinder Using LES and DES," *J. Numerical Analysis, Industrial and Applied Math.*, vol. 3, pp. 231-247, 2008.

Victor Matvienko received the diploma in applied mathematics in the Far Eastern National University, Russia, in 2007, the master's degree in visual computing from the University of Saarland in 2011, and is currently working toward the PhD degree at the University of Saarland. He is a researcher in the Interactive Visualization and Data Analysis Group, headed by Jens Krüger. In 2008-2009, he was employed in the Fraunhofer Institute for nondestructive Testing. His main research interests include flow field visualization and GPGPU computing. He is a student member of the IEEE.

Jens Krüger studied the computer science at the Rheinisch-Westfälische Technische Hochschule Aachen, where he received the diploma in 2002 and the PhD degree in 2006. In 2002, he joined Professor Rüdiger Westermann's newly established Computer Graphics and Visualization Group at the Technische Universität München. He after postdoc positions in Munich and at the Scientific Computing and Imaging (SCI) Institute joined the Cluster of Excellence in 2009 to head the Interactive Visualization and Data Analysis group. In addition to his position within the Cluster, he also holds an adjunct faculty title of the University of Utah and is a principal investigator of multiple projects in the Intel Visual Computing Institute. He is a member of the IEEE.