On Performance Analysis of Optical Flow Algorithms

Daniel Kondermann , Steffen Abraham[I], Gabriel Brostow[II], Wolfgang Förstner[III], Stefan Gehrig[IV], Atsushi Imiya[V], Bernd Jähne[VI], Felix Klose, Marcus Magnor[VII], Helmut Mayer[VIII], Rudolf Mester[IX], Tomas Pajdla[X], Ralf Reulke[XI], and Henning Zimmer[XII]

Heidelberg Collaboratory for Image Processing
Interdisciplinary Center for Scientific Computing
University of Heidelberg
69120 Heidelberg, Germany
{daniel.kondermann}@iwr.uni-heidelberg.de
http://hci.iwr.uni-heidelberg.de

Abstract. Literally thousands of articles on optical flow algorithms have been published in the past thirty years. Only a small subset of the suggested algorithms have been analyzed with respect to their performance. These evaluations were based on black-box tests, mainly yielding information on the average accuracy on test-sequences with ground truth. No theoretically sound justification exists on why this approach meaningfully and/or exhaustively describes the properties of optical flow algorithms. In practice, design choices are often made based on unmotivated criteria or by trial and error. This article is a position paper questioning current methods in performance analysis. Without empirical results, we discuss more rigorous and theoretically sound approaches which could enable scientists and engineers alike to make sufficiently motivated design choices for a given motion estimation task. [1]

[I] Robert Bosch GmbH (Germany).
[II] University College London (United Kingdoms).
[III] Bonn University (Germany).
[IV] Daimler AG (Germany).
[V] Chiba University (Japan).
[VI] Heidelberg University (Germany).
[VII] Technical University Braunschweig (Germany).
[VIII] Bundeswehr University Munich (Germany).
[IX] Linköping University (Sweden) and Goethe University, Frankfurt (Germany). With support by the Swedish ELLIIT initiative.
[X] Czech Technical University in Prague (Czech Republic).
[XI] Humbold University Berlin (Germany).
[XII] Saarland University (Germany).

[1] This article summarizes the author's results of working group discussions at the Dagstuhl Symposium on Outdoor and Large Scale, Real-World Scene Analysis held in 2011.

1 Introduction

The aim of optical flow (OF) algorithms is to compute a motion vector field based on an image sequence (the problem of defining OF properly is discussed in Section 2). OF analysis in image processing and computer vision is a comparatively young field of research with an approximate birthday in the early 80ies [1, 2]. Nonetheless, for more than thirty years, many solutions for OF problems have been proposed: a search on Google Scholar reveals that about every ten years the number of existing publications with the term "optical flow" appearing in the title doubled, reaching around 3000 this year (cf. Figure 1)[2]. Among these articles, around 150 have been published in four major journals (IJCV, PAMI, IP, CVIU) since 1980. Counting the number of publications in these journals using the term "optical flow" in the full text, the number for these journals goes up to around 1600.

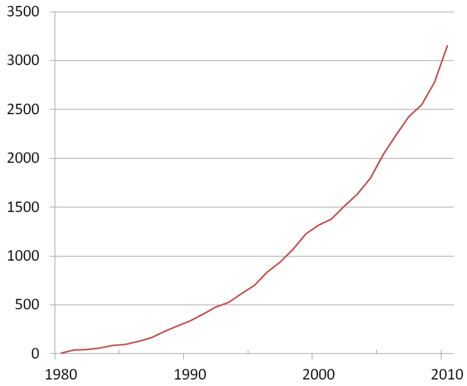

Fig. 1. Cumulative number of publications with optic or optical flow in title based on scholar.google.com (no patents, articles only in the fields "Engineering, Computer Science, and Mathematics", these fields are defined by Google).

A lot of the investigations in these papers deal with the question whether a problem for a specific application can be solved at all with image processing techniques. Today, it seems likely that many interesting problems might be solved using image processing. Although we focus on OF estimation methods, this discussion also relates to other image processing and computer vision methods such as stereo estimation, medical registration, segmentation and denoising.

Yet, with the advent of commercial applications and a ripening field of research, new challenges arise. In this position paper, we specifically discuss the problem of performance analysis which is becoming more and more important in applications such as those involving security risks (e.g. driver assistance systems). We use the term *performance analysis* rather than *benchmarking*, *evaluation* or

[2] Source: `scholar.google.com`, 26.07.2011

ranking with the intent to draw attention to the fact that the performance of an algorithm consists of a set of criteria (or requirements) that can vary with the needs of different applications and types of data. As we will discuss, we want to emphasize that performance characteristics of an algorithm cannot be described by a single scalar value.

Starting out from a discussion of contemporary performance analysis approaches in OF problems, we will address each challenge in performance analysis in a separate subsection of this text. Our aim is not to define a new paradigm for performance analysis for OF problems. Neither do the authors offer experimental results on or implementations of existing methods. Instead, the aim of the paper is:

- to review related literature,
- to create awareness for new problems arising due to the increasing number and complexity of existing OF algorithms,
- to show current trends of ongoing discussions among scientists as well as practitioners,
- to propose various new ways to characterize computer vision algorithms,
- and thereby to suggest new fields of research addressing the problems identified in these discussions.

1.1 Related Work

Both experimental and theoretical performance analysis of algorithms have a long-standing history in computer science and mathematics (e.g. rooted in complexity theory), whereas system characterization and specification is a similar strand of research in engineering (e.g. requirements analysis in software engineering).

Although many OF algorithms have been suggested, only four publications on their performance analysis exist. Chronologically, the first ones date back to 1994 [3, 4]. At this point around 500 papers with optical flow in their title had been published. In 2001, McCane et al. [5] created a new benchmark, including new synthetic scenes and a free software framework to generate new datasets. The most influential paper was published in 2007 by Simon Baker et al. [6, 7]. The authors not only created new datasets (with extraordinary efforts) and evaluated a new set of algorithms; they also created a website known as Middlebury-Database which has since been used by authors of new OF algorithms to compare their results with others. Today, around forty algorithms have been added to this database. However, compared to the very large corpus of existing work in this field, the number of evaluations is still small and lacks a theoretically justified framework.

The remainder of this section deals with papers on general theoretical approaches to performance analysis in computer vision. In later sections, we will address related work on each of the more specific topics we believe to be relevant for performance analysis of optical flow methods.

In the late 90ies, a number of workshops have been held dealing with performance analysis for computer vision in general [8–10], laying out a roadmap on why and how this strand of research should and could be established in the community. A general discussion of ten pros and cons for performance analysis in image processing was listed by Förstner [11]. This paper very much reflects the facts that on the one hand performance analysis can be very difficult, expensive and cumbersome, but on the other hand, it is also very important and feasible in terms of longterm research goals. In the same workshop, Maimone and Shafer [12] state six steps necessary for performance analysis: mathematical analysis, simulations without noise, simulations with noise, empirical testing with real data with full control, empirical testing with real data with partial control and empirical testing with uncontrolled data. A year later, these steps have been cited in a workshop editorial by Christensen et al. [8]. In 1998, Matei [13] addressed the first step by suggesting a statistical framework he called "resampling paradigm", whereas Klausmann et al. [14] concentrated on the practical question on how to evaluate performance based on given applications. They were the first to explicitly state that performance characterization and algorithm ranking are two different tasks which should be addressed only if a clear definition of the application of an algorithm is given. Therefore, they define a requirement profile and an assessment function respectively. They argue that: *"The assessment of computer vision algorithms is more than just a question of statistical analysis of algorithm results. Rather, the algorithm field of application has to be taken into account as well."*

In 2001, Courtney and Thacker [15] stated that current research focuses too much on innovation and sophistication and that performance analysis is not carried out in a well-motivated, rigorous manner. They explicitly mention that showing results on a few test images is insufficient, because it does not allow a statistical analysis. They further argue that computer vision should strictly be regarded as a branch of applied statistics. To carry out performance analyses their approach is to distinguish three evaluation types: Technology evaluation (groups of generic algorithms for generic applications), scenario evaluations (specific algorithms for specific applications) and operational evaluations (analysis of the full end user system). In a series of later papers the authors refine these ideas and suggest more concrete methods on computer vision system design [16–18].

Luxen [19] suggests to accumulate large amounts of data such as many views of the same object to achieve low errors. The results can then be used as almost noise-free ground truth. He also suggests to carefully characterize input and output data of computer vision algorithms in order to better understand under which circumstances which output quality can be expected. Similar to [15], Luxen distinguishes four levels of abstraction in computer vision systems design: intentions (e.g. image matching), functions (e.g. least squares fitting), algorithms (e.g. matrix inversion), implementations (concrete code realizing an algorithm). He argues, that each performance characterization can be based on one of these four fields. Hence, both empirical as well as theoretical studies were needed to fully characterize a system. Finally, similar to [12] he distinguishes three types of reference data for real environments: the first type are human annotations (ground truth), the second type is defined by a pair of reference *data* (without ground truth) as well as reference *code* and the third type is defined by an arbitrary implementation of an algorithm, but predefined reference input. We will discuss the generation of reference data in Section 3.4.

Further discussions on performance analysis in general can be found in [20] and [21]. The authors of [20] argue that the whole system (including all algorithms in a processing chain) need to be understood as one large optimization problem which should be solved based on a very large reference database. In [21] two important aspects are the notion that performance metrics are subject to change over time and that ground truth is very often easy to obtain in case the problem to be solved is on such a high level that humans can simply answer yes/no-questions to create ground truth. The authors also note the interesting fact that currently document analysis, face recognition and tracking/surveillance are predominant fields with many and very detailed performance analyses being published.

Most recently, in a book draft [22], Burfoot picked up on the points of [11], but in a much more explicit way. According to the author, *"The weakness of evaluation in computer vision is strongly related to the fact that the field does not conceive of itself as an empirical science. [...] Instead [...], vision researchers see themselves as producing a suite of tools."* (p.103). Burfoot further states: *"A critical reader of the computer vision literature is often struck by the fact that different authors formulate the same problem in very different ways.[...] The cause of this ambiguity in problem definition is that computer vision has no standard formulation or parsimonious justification. [...] Vision papers are often justified by a large number of incompatible ideas. [...] They will also often include completely orthogonal practical justifications, arguing that certain low-level systems will be useful for later, high-level applications."* (p. 104).

He also sees similarities to historical problems in other fields of science such as physics and chemistry: *"It is almost as if, by viewing birds, researchers of an earlier age anticipated the arrival of artificial flight, and proposed to pave the way to that application by developing artificial feathers."*(p. 106) *"The argument of this book, then, is that the conceptual obstacle hindering progress in computer vision is simply a reincarnation of one that so long delayed the development of physics and chemistry."* (p. 108) *'"The difference is that physicists can eventually determine which explanation is the best. One crucial aspect of the success of the field of physics is that physicists are able to build on top of their predecessors' work."* (p. 105)

We would like to encourage a discussion on these hypotheses with respect to optical flow estimation. In the remainder of this work we will first review what is actually meant by the term "optical flow" (Section 2). Then, we suggest a number of approaches to consolidate optical flow estimation research in the future.

2 Defining Optical Flow

Before we can characterize the properties of an algorithm, we need to clearly define what we mean by "optical flow algorithm". Using the notion of a function signature in programming, we therefore ask for input and output *datatypes*. Several definitions can be found in textbooks (e.g. [23–25]). According to Burton and Radford [23], the term "optical flow" is defined as: "the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.". This defines the output datatype to a certain degree. Remaining questions are for example whether dense or sparse flow fields need to be found; in case the actual vision system is interested in segmenting an image, the motion contours might be of interest. For tracking applications, the 3D motion of a physical object computed from the flow field could be the output whereas for motion detection, a thresholded flow field might suffice.

The question for the input datatype is more difficult to answer due to several reasons.

First, there often is no notion about the kind of images used as input. Sometimes images come from different spectra (e.g. infrared, x-ray, ...) or optical systems (e.g. fisheye lenses, omnidirectional cameras) and sometimes not all pixels in the image contain useful information (e.g. in the case of particle image velocimetry as defined in [26]). Second, mostly two images are assumed as input, therefore forbidding the use of more than two images in a sequence. Additionally, depending on how strictly this definition is interpreted, it implicitly assumes that there is a bijection, mapping pixel locations in the first image to locations in the second image. Thus, on a discrete grid, occlusions, divergences and convergences are assumed to be negligible, leaving only globally constant translations and rotations as possible outcome of optical flow algorithms.

Of course these definitions are refined or varied in each publication accordingly to describe challenges given a specific application. Usually, all approaches are subsumed under some general term such as optical flow, medical registration, stereo estimation or particle image velocimetry. This is useful to group subsets of OF algorithms with respect to their application domain and typical model assumptions. However, this terminology comes with two disadvantages: first, it is often unclear which application domains are associated with one of these groups. For example, a temporally consistent, non-dense algorithm for pixel-accurate estimation for motion utilizing more than two image frames of a sequence at once can be considered an optical flow algorithm. On the other hand, the algorithm cannot easily be compared by means of the Middlebury database for optical flow evaluation because the number of frames of the test sequences might be too small to yield good results.

The second disadvantage is that it creates the illusion in the mind of the reader, that those algorithms are comparable in their general performance. For example, an algorithm estimating motion in image sequences recorded from inside a car in order to ultimately assist the driver in detecting potential obstacles might be highly similar to an algorithm estimating the motion of a swarm of bees in their nest in order to ultimately understand the communication encoded in their dance. Yet, each algorithm can be based on completely different assumptions such as there is a planar street on which the camera is moved or that the bees move on a hexagonal grid. The algorithms might also address different problems as for example the occlusion and translucency of cars or motion blur of the bee's shaking bodies. Furthermore, the outcome of the algorithm might be subject to requirements such as sub-pixel accuracy for time-to-impact computation versus good motion boundaries for bee-body segmentation.

Due to these disadvantages of adding all OF algorithms to a single group, we believe that a very careful categorization based on the properties defined in the following sections is crucial for further advancements in the field.

As correspondence problems are mostly ill-posed, prior knowledge about the estimates to be computed is always needed. This knowledge should be well-understood and described as well as possible and also be as accurate as needed for the task. On the other hand, it should generalize well over many types of input data. Therefore, as in machine learning, a trade off between generalization and specialization for the model needs to be found. This condenses to the question: which model is too general and which is too specific? In contrast, current approaches to performance analysis try to categorize existing algorithms either based on the employed optimization framework (e.g. local versus global and variational versus graphical models) or are based on a single scalar output criterion such as the average endpoint error (defined as $E_{ep}(\boldsymbol{x}) = ||\boldsymbol{u(x)} - \boldsymbol{g(x)}||_2$ with $\boldsymbol{x}$ being a pixel location, $\boldsymbol{u}(\boldsymbol{x}) = (u_x(\boldsymbol{x}), u_y(\boldsymbol{x}))^T$ the computed flow and $\boldsymbol{g}(\boldsymbol{x})$ the true flow, respectively). Yet, instead of being fixed to a single criterion such a ranking needs to take into account all requirements of a given application.

Given a system that uses correspondences as input data (an application), requirements analysis (cf. e.g. [27]) helps to understand how specific the model can be without loss of generalization within the bounds of the application domain. But working with requirements implies knowledge about the application domain. Hence, in order to clearly define OF algorithms we need to create a categorization of applications, which will be discussed now.

2.1 Application Categorization/Systematization

In order analyze the appropriateness of a model for a given application, we need to know the application. On the other hand, there might be an infinite number of yet unknown applications for OF algorithms. It seems unlikely that we can first enumerate all applications and then analyze the performance of each and every algorithm for each and every application. System engineers (cf. e.g. [28]) found a way around this problem by identifying a number of meaningful and intuitive properties for each system component which are measured and then listed in a specification sheet. These properties are selected by finding those which are, ideally, important for as many relevant applications as possible. In order to select the most indicative properties, all currently available applications are considered. Then, by experimentation, system properties are selected and tested for their usefulness.

Currently, the two most important properties for OF algorithms seem to be average endpoint error (disparity error in stereo) and (to some extent) algorithm complexity (computation time, memory efficiency). By looking at some well-known applications we will see that there is a variety of other properties that are important and sometimes contradicting each other.

For photogrammetry and 3D reconstruction [29, 30] correspondences are the basis for triangulations: if the 2D position of the same 3D point is known for two or more views in a number of images, the 3D position can be reconstructed via projective geometry. The accuracy of this reconstruction largely depends on the accuracy of the correspondence (which, in turn, depends on system configuration parameters such as camera baseline, etc.; cf. e.g. [31]). Such methods typically require a large number of correspondences which are not spatially correlated by regularization techniques. The correlation due to these (in general necessary) techniques is a severe problem in statistical analysis as it is difficult to characterize. If the regularization is data dependent or robust estimators are used, the problem becomes even more theoretically involved.

As soon as very large scenes have to be reconstructed, speed and memory efficiency become an issue as well [32]. Here, a tradeoff between speed and accuracy has to be found. This leads to the notion of "scalable algorithms" where an optimum tradeoff can be found by adjusting system parameters.

In robotics and driver assistance systems, OF algorithms have different requirements: in this scenario the task often is to merely detect objects such as traffic signs, the ground plane or sources of danger. Here, speed, memory and energy consumption play a crucial role. On the other hand, sparse flow fields often are sufficient, e.g. for navigation and localization [33, 34].

Correspondences are also used to interpolate intermediate frames between two consecutive time steps of an image sequence [7, 35]. A related case is stereo baseline adjustment or, more general, view synthesis based on multiple images [36]. Software companies involved in cinematic movie postproduction such as The Foundry (Nuke) implement a number of (modified) methods known from literature but are not always published [37, 38]. In these applications the correspondences need not necessarily be physically correct; the most important property often is is that they are temporally consistent and can be used to produce results which are pleasing to the eye.

The opposite is the case in scientific measurements. Application scenarios are for example the mensuration of water waves or plant growth in environmental physics [39, 40], estimation of air streams around objects [41], weather-forecasting [42] and the analysis of fluid motion in heart-assist devices [43]. In all these cases, a small endpoint error of the flow vectors has the highest priority, whereas speed often plays a minor role. Furthermore, the confidence (cf. Section 3.2) of each individual measurement needs to be estimated to allow researchers to asses the outcome of each experiment. Interestingly, for these applications, completely parallel fields of research with little overlap to image processing or computer vision have been established [26, 44], bringing up similar concepts of correspondence estimation, but focusing on different approaches (e.g. block-matching for motion estimation [45]).

A number of other fields of research deal with optical flow, such as action recognition [46], video surveillance [47], video compression [48], video annotation [49], supervision of elderly people [50], swarm analysis of beehives [51] as well as research in zebrafish embryo development [52].

The abundance of existing and possible applications indicates that a complete overview of applications is difficult to define and maintain. On the other hand, based on the requirements of a subset of these applications, a set of more abstract algorithms properties could be found. Similar to specification sheets of electronic system components, we believe that OF algorithms can be described by carefully characterizing input and output data as well as system properties.

Once a definition (or a set of definitions) for algorithms has been found based on applications and their requirements, we would like to understand how well a given algorithm performs. To answer this question, several challenges have to be solved. This will be discussed in the following section.

3 Challenges in Performance Analysis

We identified five points to be considered to thoroughly characterize the performance of an algorithm:

- Input data characterization can help to organize typical image sequences into categories with similar properties (Section 3.1).
- Output data characterization should not only evaluate the accuracy of OF methods. Instead, we suggest a list of six important properties of output data (Section 3.2).
- System properties describe the technical aspects of speed and memory consumption as well as modularity and engineerability (Section 3.3).
- The problem of ground truth generation is largely unsolved, but is one of the most crucial as well as difficult aspects of performance analysis for OF algorithms (Section 3.4).
- Finally, well-motivated performance metrics for the comparison of flow fields have to be found (Section 3.5).

Each of these points is carefully motivated in the following subsections. Related work will be discussed along with suggestions how each topic can contribute to a more thorough and theoretically motivated approach to OF performance analysis. In Section 4, we will discuss hypotheses why so few performance analyses for OF are currently available and why a detailed consideration of each of our points could boost both quality and quantity of optical flow research.

3.1 Input Data Characterization

As discussed above, the type of data inserted into an OF algorithm is not always sufficiently described as ”image pair”.

Qualitative Characterization First steps in input characterization could be to describe the image acquisition process and the content of the scenes for which the algorithm should work. In many specialized publications as for example in medical image registration the mode of data (x-ray, ultrasound, ...) is usually defined clearly. To extend this description of input data it would be beneficial to describe the full imaging setup including sensors, lenses, lens settings (numerical aperture and focal length), light sources (incident angle, physical shape, spectra), surface material properties (reflectance functions), etc. This is usually done in particle image velocimetry were the setups vary largely [26]: in this special case the input data is a 2D image generated laser sheet that visualizes particles. Here, the motion is considered to be truly 2D-dimensional so that apparent flow and physical motion coincide.

Describing and categorizing the acquisition process and the content of the scenes creates awareness for the task the algorithm was made for, but it will often be difficult to exhaustively explore the data when the algorithm is supposed to work well and when not. Another way to solve this challenge might be the analysis of large amounts of input data ideally fully describing inputs which are suitable for the algorithm.

Quantitative Characterization Local feature vectors containing e.g. orientation and scale information could be used to decide whether a given scene is similar enough to yield acceptable results with the OF method at hand. It might be useful if these features were directly related to known critical situations such as occlusions, low amounts of texture, illumination changes or large motions. Also global features describing the image or the scene as a whole and comparing it to sequences with known outcome might characterize input data in a useful way. However, it remains to be studied whether purely local or purely global features can express the full complexity of data sufficiently for a given application.

There are several possibilities to characterize the specific set of image sequences which are addressed by an OF algorithm. First of all, much research has been dedicated to scene descriptors (e.g. GIST is popular approach [53]). Another possibility is to characterize the structure of the (single) images by more or less standard techniques, such as describing the spatial autocovariance function; this can be done compactly by setting up parameterized models, such as separable exponential decay functions. This description should be completed by at least a rough description of the noise variance. A more careful and detailed characterization would include a parameterized description of the optical point spread function as well as the spatial sensor element dimensions (fill factor, or more detailed). The overall characterization of the *discrete* inter-pixel autocovariance results then from convolving the optical and sensor characterization, and the intrinsic autocovariance function of the image, as it would be if the former two influences were neglectable. This intrinsic image autocovariance function corresponds to what is often discussed as the 'natural' and ubiquitous power spectrum of images *per se*, often modeled as an $1/f$ power spectrum.

The *temporal* characterization should consider the exposure time(which can range between a small fraction of the temporal distance of frames, and the full inter-frame period). More importantly, the temporal characterization should describe the distribution of apparent 2D velocities (or displacements). In the case of certain applications, in particular for driver assistance, this distribution can be significantly different across the image area, and it can also be dependent on some (measurable) external parameters such as camera motion w.r.t. the fixed world coordinate frame. These characterizations do of course not capture the full characteristics of an 'interesting' image sequence, which is structured into differently moving objects, has occlusions, etc., but it is already a very solid basis for optimally designing the derivative operators needed for all differential methods [54] for designing averaging operators (instead of resorting to 'Gaussians') [55], and furthermore to provide useful priors for the entities which are sought.

In an ideal scenario a set of generative input data models (acquired e.g. by machine learning) could be found which can reliably be used to describe the input data the algorithm was made for. As will be discussed in Section 3.2, another intriguing aspect of input data characterization is to identify local regions in a scene were the model cannot be applied to because it is either too specific or unspecific.

3.2 Output Data Characterization

As the results of OF methods are used for many different applications, the quality with respect to a given application can be defined with various optimization goals. Hence, next to characterizing input data the same should be done for the resulting flow fields. We will now describe several approaches starting out from very basic characterization techniques such as using example outputs and qualitative evaluations. Then, we will shortly discuss two seldom addressed output data properties, namely robustness with respect to model violations as well as temporal consistency of flow fields. Finally, we will review research on the heavily studied question for accuracy and a currently evolving approach to confidence estimation.

Example Output The most basic and also a very general way to characterize output data is to provide example outputs of the algorithm. This can for example help programmers to check the correctness of a reimplementation of the method at hand. If large amounts of results are available on various kinds of data it can also facilitate the choice of algorithm for a specific application.

Qualitative Evaluations Another basic approach are qualitative evaluations. In creative image processing, aspects such as visualization, rendering and post-processing of videos, the mere beauty of the results can be of major importance. Typical cases are frame interpolation as well as view synthesis. In such cases it might also be possible to "cheat" on the viewer by creating false results which have no noticeable effect on the outcome of the application. These scenarios also allow for psychological tests analyzing whether the viewer is able to find the errors in, or is otherwise affected by the algorithm outcome. We did not find related work on psychological studies for these approaches in the field of OF research.

Robustness With Respect to Model Violations In safety-relevant applications such as driver assistance and medical systems, the robustness of model and optimization strategy with respect to data violating the model is of great interest. As there is an infinite number of possible model violations it is difficult to devise general tests. One way to describe output data with respect to model violations is to collect large amounts of data containing common model violations, such as motion blur, lens flares, etc. Another closely related question is how fast the results deteriorate if the model is violated. In case the quality degrades gracefully, the algorithm might be better suited for those applications dealing with safety issues.

Temporal Consistency For video processing, the temporal consistency of the algorithm results are often more important than other properties. A test for this consistency could be carried out by systematically varying original data to see how the outcome changes. This is similar to sensitivity analysis in linear models [56] and machine learning approaches. Two recent articles enforcing this property and showing very promising results are [57, 58].

Accuracy Limits There are several ways to test and compare accuracies of OF algorithms. A major problem is how to measure the error because there is an infinite number of options to define an order (or ranking) between two vectors. Hence, each pair of vectors (i.e. ground truth and measured flow vector) first has to be transformed into scalar values in order to be comparable. Next to the regularly used endpoint error [7] various choices exist. One way is to compute the magnitude of both vectors. This is problematic when ground truth vector and measured flow vector are on the one hand equally long but on the other hand point into opposite directions. The magnitude error would still be zero. Another way would be to compute the angle between two vectors which raises the analog problem: The vectors can be of different magnitude. Another problem here is the singularity for vectors of very small magnitude. To weight these two components of magnitude and angle the so-called angular error defined by [59] has been suggested. This error weights both parts of the errors in a nonlinear and unintuitive manner which was not motivated in the original paper (as discussed in [60]). Depending on the application one error measure or another might be favorable, a fact that should be taken into account when stating the accuracy limits of the algorithm.

Once an error measure has been defined, the error distribution needs to be sufficiently motivated. The problem here is, that this distribution actually depends on image data, ground truth and measured flow. For example, testing of the accuracy with a highly textured region that moves at a constant velocity everywhere yields very low errors with most algorithms. If the images were of constant color (one homogeneous region) the results could be completely wrong. The ground truth could also be arbitrary. Hence, testing on a sequence like Yosemite [61] (or any other small set of sequences) does not adequately represent the quality of the algorithm. It just gives a hint that for this type of scene (e.g. highly textured, smooth and mostly small motion in case of the Yosemite sequence) the algorithm might actually work well. Thus, in some cases algorithms work very well for extremely small motions, sometimes for very large motions. These limits should be well understood and clearly stated.

Furthermore, representing the error distribution only by its mean and variance for a full image is not sufficient, because only the Gaussian distribution can be fully described by these first two moments. As motion estimation errors are far from being Gaussian distributed it might be more helpful to actually visualize the whole distribution (or parts of it) which in turn raises the problem of density estimation. Another option could be to show per-pixel error measures as is done on the Middlebury website.

Finally, it would be helpful if it was known under which circumstances the most accurate results can be achieved by an algorithm. At first this sounds easy to answer: Constant motion through time and much texture certainly is a simple case. Yet, an image of a Gaussian intensity distribution in a 32 bit quantized image might even yield very accurate results for non-constant motions such as a rotation. Furthermore, it is interesting to which degree the results deteriorate with respect to more challenging image data. To the best of our knowledge, this aspect has never been studied thoroughly although it is very important for scientific applications where sub pixel accuracy is critical and where it is safe to make more specific assumptions about the model.

Estimatibility, Confidence and Alternative Solutions Usually, OF algorithms are analyzed by comparing ground truth with actual algorithm results. This type of performance analysis is carried out by humans prior to the actual usage of the algorithm in a full computer vision system. Therefore, we call this technique *supervised performance analysis*. An alternative approach is to allow for self-diagnosis of the computer vision system while it is running in its real environment. We call this approach *unsupervised performance analysis* which will be described now.

To motivate three aspects of unsupervised performance analysis consider the following extreme example of OF input data: A typical image sequence for particle velocimetry consists of a mainly black background and some hundred (or thousand) bright moving spots which are physical tracer particles in a fluid. In the black (homogeneous) regions of the background no motion can be estimated: a black spot at any location can be matched to almost any other location in the next frame. We do not care about these occlusions and ambiguities in the background and simply assume that there is no motion at all. Hence, an algorithm working on this data should be able to decide where motion can (or should) be estimated *at all.* Furthermore, particle velocimetry is often used in environmental sciences to measure fluid motion, so each and every measurement must come with (at least) an error bar, showing the *precision* of each flow vector. Finally, occlusion occurs whenever two particles are crossing due to the projective nature of the image acquisition. Sometimes, it is impossible to decide which particle is which after they crossed in the image plane. Therefore, our algorithm needs to be aware of *alternative solutions.*

More generally, we ask how much information we *need* to obtain from the given data and how much we *can* obtain depending on the intended later use of the resulting motion:

- Dealing with occlusions and ambiguities can be understood as dealing with estimatibility: Instead of assuming that at each pixel of an image sequence a full flow can be estimated, we pose the question whether motion can be estimated at all and, if so, how many parameters of it [62, 63]. This should be easier to decide than to actually carry out the estimation.
- To answer the question how accurate the results are we use confidence measures. This should still be easier than computing an actual flow field.
- Finally, the most algorithmically complex and related task would be to not only find one motion estimate but to also inform the user about alternative solutions.

These notions of estimatibility, confidence and alternative solutions also relax the problem of motion estimation: we do no longer need to estimate flows at each and every pixel. This reduces both computational cost and potentially harmful results in safety-relevant applications such as driver assistance systems.

While little literature focuses on estimatibility and alternative solutions for optical flow, confidence measures have already been studied by Barron et al. [3]. A first paper specifically dedicated to the comparison of confidence estimation approaches has been published by Bainbridge and Lane in 1996 [64].

Two approaches are regularly being studied: confidence based on input data (images) and confidence based on output data (flows). As the first does not take the results into account, they can also be interpreted as estimatibility measures. A central theme recurring in all image-based confidence methods is the notion of the local shape of the energy to be minimized. The intuition is that sharp peaks in the energy indicate high confidence whereas low curvatures allow for many equally likely flows resulting in a low confidence.

More formally, two highly related theoretical frameworks can be used to describe this approach: intrinsic dimensions and Fisher information (both defined e.g. in [65]). Both definitions are based on the local covariance matrix of the energy of an OF model. Intrinsic dimensions can for example be used to determine the number of parameters which can be estimated [62]. They have firstly been applied in computer vision in 1990 [66] and later been adopted e.g. in [24] and [67, 68]. Fisher information is used to describe the Cramér-Rao Lower Bound which states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information. Therefore, this bound is an indicator of how accurate the best possible outcome of the motion estimate can be. Another option is to use the Chi-Square-Test which can be used to verify the appropriateness of a model in case the errors are normally distributed, unbiased and have a given assumed covariance matrix.

A different way to estimate confidences is to solely rely on prior knowledge on flow field statistics. This has been studied for example in [69, 70], where the spatio-temporal statistics of typical flow fields are learned in terms of a linear model which is then used to employ hypothesis testing on OF algorithm results. Similar approaches on learning the statistics of flow fields have previously been applied to OF estimation (e.g. in [71–73]).

Two recent publications [74, 75] use learning based on multiple clues derived from both image and flow data for confidence estimation.

Finally, scene-inherent redundancy could be another aspect for confidence estimation: in case one has three or more images, the results should be consistent with respect to to the geometry of the scene, e.g. rays to the same scene points should intersect. This goes beyond the Fisher information, as additional flow fields of other pairs of images of a static scene can be used to define a local flow vector quality criterion.

3.3 System Properties

Until now we have focused on the algorithm definition as well as the input and output data characteristics. All these properties focus on the data an algorithm receives and computes. Another important point is to understand all relevant technical details of concrete implementations. Hence, a set of system properties needs to be found so that engineers can deal with a system to compute flow fields as black box. We identified three major groups of such properties: the ease of maintenance and implementation, the possibility of white-box testing and speed as well as memory usage.

Engineerability and Number of Parameters We understand engineerability as the ease of implementation, the possibility to actually implement the algorithm in a commercial application and the possibilities of adapting the method to the specific needs of engineers. Especially the number of parameters influencing the output of the algorithm should be small in their number, intuitive to understand and insensitive with respect to input data. In case the number of parameters cannot easily be reduced, a set of default values should be known which can be used to create results of reasonable quality on most images. Commercial aspects such as whether the algorithm is patented or not might also play a role. This system property can be tested easily by explaining and motivating the parameters thoroughly and estimating the amount of time a programmer new to the field might need to implement the method.

Modularity and White Box Testing A common practice in the publication of OF algorithms is to describe the whole algorithm and to test its output against test sequences. Regularly, a few crucial parts of the algorithm are either left out or parameterized differently in order to estimate its effect on the overall results. For example, many OF algorithms are built up from many algorithmic elements, such as multiple similarity measures, image derivative kernels, interpolation techniques, pyramid computation schemes, regularization terms and so on. Each of these elements has parameters and can even be replaced by completely different methods. For example, sub pixel image intensities can be interpolated by a number of interpolation schemes; an image pyramid can be computed by scaling the original image down by a factor of two or smaller or it can even scale the image up to some degree [76]; the derivative of an image can be computed by many kernels or even other filtering techniques ranging from simple central differences to sophisticated filters specially designed to estimate motion with a specific similarity measure [77]. Any subtle change in these settings can influence the overall accuracy of the results and is therefore worth further investigation.

At the core of this problem lies the fact that any OF algorithm is actually plugged together from a large set of modules available. Some of these modules as for example image derivative computation are fields of research on their own. It would be helpful if there were a set of known slots (constituting the elements of the most general motion estimation algorithm and clearly defining input and output data) and a variety of possible modules that could be plugged into each appropriate slot. Then, each slot or module could be scientifically investigated separately and also in its combination with other modules (white box testing). A software framework for this approach including a number of example optical flow algorithms has recently been made publicly available[3]. The software is based on a modularization strategy specifically designed for OF algorithms as suggested in [78]. These modules of an optical flow method are another interesting set of algorithm properties.

Execution Speed and Memory Usage The time and memory an algorithm needs to actually estimate the motion of an image sequence usually is a major issue in industrial applications. Several aspects range from practical over completely theoretical to technically highly intricate considerations; to each of these a complete field of research is dedicated. Therefore, it is very difficult to judge the execution speed of an algorithm even though it is one if its important properties.

- Data Reduction: Sometimes, it suffices to only compute motion at a few locations. Hence, computation time can be saved by finding algorithms that reduce the number of locations. This is a typical approach in tracking [79] where usually only very few pixels of an image sequence are investigated.
- Mathematics: For example in global motion estimation techniques (often including systems of partial differential equations), large linear systems of equations are generated from the image sequence. Their solution can be carried out by many methods, ranging from Gaussian Elimination Schemes over Krylov Subspace Methods to Algebraic Multigrid Schemes. Exploiting

[3] http://charon-suite.sourceforge.net

mathematical properties can dramatically reduce computation times. This was for example shown by [80, 81].

- Parallelization: With the dawn of multicore desktop computers and general purpose GPUs, parallelization has become a major topic. Especially in image processing, parallelization is surprisingly easy to implement (consider e.g. the convolution of an image with a mask). But also solving large linear systems of equations can be done in parallel (cf. e.g. [82].
- Code Optimization: It might sound trivial but with a diversity of large image processing libraries for major programming languages (as e.g. C++ and Matlab) code optimization is far from simple. Nonetheless, this part can also affect theoretical considerations: if it were for example easier to optimize code for matrices than for other data structures such as graphs, the choice of the optimization method would interfere with the actual code design. Today, a programmer needs to have a deeper understanding on how image processing libraries implement their functionality in order to optimally exploit its internal structures. Another problem is that the ways compilers optimize code is rather unintuitive: one cannot implement all functions in the same way to yield the same automatic code optimizations. A typical approach is trial and error, but each compiler optimizes its code differently so that the same code can be much faster when compiled with a different compiler.
- Choice of Hardware: For some methods, specifically designed hardware ranging from image acquisition device to integrated circuits for numerical optimization can influence the execution speed. For example, modern driver assistance systems contain integrated modules for stereo estimation which deliver highly accurate depth maps in real time with very low power consumption. Another example are highly optimized detectors in the large hadron collider which can detect and transfer collisions in the gigabyte-range per second. Finally, the famous Microsoft Kinect creates depth maps in real time with a customized hardware setup for structured light. This shows that a focus on regular personal computers is not necessarily the best way to decide whether an algorithm can be fast or whether some specific problem can be considered as solved.

Hence, investigations into the various complexities of optical flow algorithms are an important property to be specified.

3.4 Ground Truth Generation

The typical approach to evaluate the quality of output data is to design ground truth image sequences where the motion is known. Two approaches can be chosen:

1. Synthetic image sequences are generated. Due to the underlying and known 3D models, the true motion field is generated easily from animation data. The problem with this approach is that rendered images can be unrealistic. In fact, it is unknown whether renderings are realistic enough to fake real-world scenes.

2. Real images are recorded. The motion is measured by some technique which is more accurate than optical flow methods. The problem with this approach is that the measurement motion can be inaccurate and that very few accurate motion estimation techniques are known. This leads to scenes with limited content such as scenes with rigid body transformations, small sets of a collection of rather artificial items and the like.

The dilemma in ground truth generation therefore is that either the ground truth flow fields are too inaccurate or the recorded image sequences are too artificial. The most famous examples for synthetic scenes are the Yosemite sequence [61], the street and office sequences [5] and the diverging tree sequence [3]. Of course, they do not cover all types of applications and can therefore only be used as a hint on how the algorithm might perform on other sequences. One problem with such sequences is that it is largely unknown whether they represent important or typical cases of motion together with the rendered images. Furthermore, there are sequences which are acquired with a real camera. The first well-known example is the marbled block sequence [4] which contains a few block-shaped, textured objects standing on a textured underground. Recently, a number of new synthetic and real sequences have been generated by [7]. Furthermore, the authors of [7] encourage the publication of results based on a website were everyone can submit new motion fields. For automotive scenarios two large datasets have been published [83, 84]. They both contain very large amounts of representative data, but for [83], no ground truth is available and for [84] only parts have been augmented with ground truth.

Furthermore, the generation of ground truth data is a challenging optical measurement task itself. Its accuracy should ideally be magnitudes above the accuracy that can be achieved by motion estimation algorithms. The typical problem of real sequences is the estimation of this accuracy. In the publications mentioned above the information supplied from an optical measurement perspective seems to be insufficient to clearly state accuracy limits. Hence, even though in real sequences all physical imaging effects from lens distortion and noise to light reflections and refractions are modeled properly, it remains unclear whether their ground truth is good enough. In such circumstances, when ground truth of real world data is either difficult or impossible to obtain, one can either use human-assisted motion annotations [85, 86] and carefully evaluate the accuracy of the resulting flow fields or one can try to synthetically create image sequences with known ground truth. One tool to achieve the latter has recently been suggested by [74].

Then, an open question is whether rendered scenes are sufficient to simulate the real world with respect to OF methods. Inspired by a first analysis of real versus synthetic data [84], in [87, 88] the goal was to create the same scene both in the computer and in reality and to compare the outcome of a given OF algorithm. In case the two results do not differ significantly, we can conclude that computer graphics can be used to simulate at least a part of reality. How large this part is would then be subject of further investigations. Yet, along with [11], we would like to stress the point that simulated data are absolutely necessary to prove the correctness and potential accuracy of algorithms.

Finally, the selection of the (ideally) *best* datasets is a big and completely unsolved problem. In practical applications, we are required to evaluate OF without ground truth. Therefore, we need to believe that the results computed by algorithms which derive acceptable performance for reference data are also acceptable for other sequences. However, to accept this meta-criterion, we are required to accept the pre-assumption that mathematical, geometrical, and physical properties of the test data are at least comparable to the previously used reference datasets. Therefore, we need to evaluate the quality of our datasets: which scenes do represent real data best for a given application?

In current real datasets, the camera often does not move. In synthetic datasets the camera is often flying smoothly through the scene. Both types of camera motion seem unlikely in real-world situations such as robotics or driver assistance systems. On the other hand, in surveillance applications a static camera can very often be assumed, whereas in airborne settings, a smoothly flying camera might be a good assumption. Next to the camera motion, the content of the scene and the motions present in the scene have to be decided on some well-motivated thoughts. For example, most probably nobody wants to estimate the motion of fireworks exploding in a breaking ocean wave during a blizzard with big snow flakes and lightning bolts. On the other hand, difficult sequences such as the motion analysis of a soccer game during rain with hundreds of strobe lights triggered by reporters can be highly valuable.

Another problem for the best selection of sequences with realistic camera and object motions is the length of the sequence: In case motion is temporally coherent in our reference datasets, we can use the computed results of the previous frame as a guide to evaluate the results in the present frame. Yet, this property implies that to evaluate algorithms which will be applied to long image sequences, we are required to prepare reference image sequences which satisfy the same temporal motion coherence along the time axis.

Little related work on this topic exists; a first step towards the question of good datasets was proposed in [89]. Outside the field of OF, Shotton et al. showed that for human pose estimation it is feasible to build a challenging synthetic test (and training) dataset [90]. Kaneva et al. used this idea for feature estimation [91].

Even if ground truth data could be easily generated in large amounts, it would still be unclear whether a generalization of the image data created across all fields or even inside each field of applications can be found. Thus, the quality assessment of something like a general-purpose optical flow algorithm might still be impossible: We would have to test it with all types of test sequences we can imagine. Therefore, even if a general-purpose algorithm were found, we would possibly never be able to identify it. We argue that to alleviate this problem many more sequences need to be created. If a generative model for input characterization methods (as discussed in Section 3.1) would be found, one way to use it would be to generate such large amounts of ground truth. As optical flow scientists usually have a specific type of images in mind, another way to alleviate this problem is to supply the source code of the algorithms in order to enable other scientists to carry out tests with their own data. This method may seem obvious but is, unfortunately, not always put into practice.

3.5 Performance Metrics

New motion estimation algorithms are usually tested with a number of ground truth sequences. Until the Middlebury Database was established in 2007, they where often solely tested against the Yosemite sequence [61]. As an error measure usually the so called average angular error (defined by [59] and used by [3] and most successive papers) and its standard deviation over a single frame of this sequence is reported. Not only is this error measure unmotivated, it also is inappropriate for the comparison of some typical problems of motion estimation as is e.g. laid out in [24]. To address these problems, an additional set of performance measures was introduced by [7]. But it is not obvious which measure can best be used to compare the estimated results to the ground truth.

To put it in a nutshell, currently used performance measures are of questionable use in real-world application scenarios. A lot of future research could be carried out in this field.

4 Conclusion and Future Research

In this paper we have discussed the importance of performance analysis for optical flow algorithms. A number of algorithm characteristics have been proposed to help scientists as well as engineers to design improved algorithms or choose between several options for a given application.

How can we facilitate systematic performance analyses of existing OF algorithms?

In the past much attention was paid to *innovation* of new methods rather than *consolidation* of existing methods. This resulted in an abundance of publications. To better understand these findings in OF research, we suggest the following first steps to consolidate existing work.

4.1 Creation of Reference Implementations.

Creating a new implementation of existing methods is a time-consuming task due to the increasing complexity in current modeling and optimization techniques. Often, much theory knowledge and programming expertise are needed. Yet, it would help to have multiple independent implementations of each OF method for performance analysis.

Implementing an existing method could be rewarded by scientific reputation: the online journal Image Processing On Line is dedicated to certifying algorithm implementations[4], so that peer-reviewed implementations of OF methods become part of a scientific result. This approach has many advantages:

[4] `www.ipol.im`

- Peer-reviewed reference implementations would be generally accepted by the community.
- Comparisons to baseline methods became possible without ambiguity due to implementation details.
- The workload of reimplementing existing methods is distributed over the community.
- Performance analyses of new methods become easier.

For future research, we encourage students and scientists to publish peer-reviewed reference implementations to create a basis for consolidation in OF research.

4.2 Creation of a System Characterization Standard

We have suggested a number of ways to characterize OF algorithms. We showed that, next to accuracy, speed and innovations in modeling, there are many interesting properties. Characterizing them could lead to new approaches with very good tradeoffs for the specialization-generalization-dilemma stated above. This article is a step towards more awareness for system characterization in OF. Further position papers, workshops or even dedicated journals or conferences could help to create a system characterization standard which is supported by a majority of researchers.

4.3 Specialization of Publications on Subtopics in OF

Historically, publishing a new paper in OF is done by reviewing the related work and describing a model as well as optimization technique. Experiments are shown indicating that the proposed method works well under reasonable assumptions. In the nineties, the number of publications was already so large that it became difficult to exhaustively describe the related work. The first review papers emerged and authors of new methods concentrated on the closest related work in order to be able to keep the page limit.

Today, models and optimization techniques become more and more sophisticated and the number of OF publications has grown out of the bounds of an exhaustive review paper. Additionally, performance analysis has become more important as engineers need to choose from among thousands of publications "the correct" method for their specific application. As a result, it became difficult to give all answers about a new approach within a single publication.

Breaking down the OF problem into parts which can be handled conveniently and in great detail within a single article could therefore be beneficial. One approach could be to only propose a new model in a baseline optimization framework and show that the results make sense (but without performance analysis) and the idea is innovative. Other researchers could create and/or use a reference implementation to study its properties as proposed in this article. Yet another group could compare the results with those of other methods. Finally, a paper about many comparisons could come to a conclusion about the question which method is most appropriate for which task.

Thus, innovation and consolidation and could be significantly facilitated.

4.4 Usage of White-Box-Testing for Performance Analysis

Black-box-testing analyzes the properties of an OF algorithm solely based on its output [92]. The advantages of this approach are that no knowledge about the internals is necessary and users will experience the same behavior. A disadvantage is that it remains unclear which component of the method caused a change of the system properties: for example, exchanging a scheme for pyramid or derivative computation can have a large impact on the output.

Whenever multiple modules are modified at the same time, black-box-testing is no longer suitable to interpret the results: it might be possible that two of three modified modules degrade the outcome whereas the third modules yields a very significant improvement. If several results from more than one publication are to be compared, we simply cannot change one module at a time.

These are reasons to use so-called white-box-testing, meaning that the effect of each module of an algorithm on the system properties should be analyzed separately. One approach is to segment OF algorithms into independent modules and create reference implementations for each module separately. Algorithms sharing several modules such as pyramid or derivative computation can then be easily compared. This approach has been described in [78] and resulted in a freely available, modular software suite called Charon [5].

4.5 Development of a Simple Ground Truth Generation Technique

Categorizing OF applications and finding a way to characterize input and output data as prerequisites for thorough performance analyses is a difficult task in its own right. Until a standard approach has been found it would still be useful to evaluate OF algorithms with respect to specific applications. Creating many ground truth sequences is a good way to achieve this, but as the number of applications is very large it is difficult to create so many sequences. Ideally, everybody should be able to easily create new ground truth satisfying some well-defined quality constraints. Possible candidates for such an approach would be synthetic image sequences or 3D scanning. Both ideas need a sound scientific validation before they can be employed as a black box.

4.6 Summary

We have suggested five directions for future research: reference implementations, system characterization standards, subtopics for publications, white-box-testing and simple ground truth generation. A better balance between consolidation and innovation could be found by these approaches. With this article we hope to inspire scientists to have a closer look at what has already been achieved in our field of research.

[5] `charon-suite.sourceforge.net`

References

1. B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.
2. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130, 1981.
3. J. L. Barron, D. J. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
4. M. Otte and H. Nagel, “Optical flow estimation: advances and comparisons,” in *Proceedings of the European Conference on Computer Vision*, pp. 51–60, 1994.
5. B. McCane, K. Novins, D. Crannitch, and B. Galvin, “On benchmarking optical flow.” http://of-eval.sourceforge.net/, 2001.
6. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” in *Proc. of the 11th International Conference of Computer Vision, (ICCV07)*, pp. 1–8, IEEE, 2007.
7. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.
8. H. Christensen and W. Förstner, “Editorial performance characteristics of vision algorithms,” *Machine Vision and Applications*, vol. 9, no. 5, pp. 215–218, 1997.
9. R. Haralick, R. Klette, S. Stiehl, and M. Viergever, “Performance characterzation in computer vision,” 2000.
10. A. Clark and P. C. (eds.), “ICVS workshop on performance characterization and benchmarking of vision systems,” 1999.
11. W. Förstner, “10 pros and cons against performance characterization of vision algorithms,” in *Proc. of ECCV Workshop on Performance Characteristics of Vision Algorithms*, pp. 13–29, 1996.
12. M. Maimone and S. Shafer, “A taxonomy for stereo computer vision experiments,” in *Proc. of ECCV Workshop on Performance Characteristics of Vision Algorithms*, pp. 59 – 79, April 1996.
13. B. Matei, P. Meer, and D. Tyler, “Performance assessment by resampling: rigid motion estimators,” in *Proc. IEEE CS Workshop on Empirical Evaluation of Computer Vision Algorithms, Santa Barbara, California*, pp. 72–95, 1998.
14. P. Klausmann, S. Fries, D. Willersinn, U. Stilla, and U. Thönnessen, “Application-oriented assessment of computer vision algorithms,” *Handbook of computer vision and applications*, vol. 3, pp. 133–152, 1999.
15. P. Courtney and N. Thacker, “Performance characterisation in computer vision: The role of statistics in testing and design,” *Imaging and Vision Systems: Theory, Assessment and Applications, NOVA Science Books*, 2001.
16. N. Thacker, A. Lacey, P. Courtney, and G. Rees, “An empirical design methodology for the construction of machine vision systems.,” *Tutorial at ECCV, Copenhagen*, 2002.
17. N. Thacker, “Using quantitative statistics for the construction of machine vision systems.,” *Proceedings of SPIE: Opto-Ireland 2002: Optical Metrology, Imaging, and Machine Vision*, vol. 4877, pp. 1–15, 2003.
18. N. Thacker, A. Clark, J. Barron, J. Ross Beveridge, P. Courtney, W. Crum, V. Ramesh, and C. Clark, “Performance characterization in computer vision: A guide to best practices,” *Computer Vision and Image Understanding*, vol. 109, no. 3, pp. 305–334, 2008.

19. M. Luxen, "Performance evaluation in natural and controlled environments applied to feature extraction procedures," in *Proc. of 2004 ISPRS Congress*, vol. XXXV, Part B3 of *The International Archives of The Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 1061–1066, 2004.
20. Y. Lucas, A. Domingues, D. Driouchi, and S. Treuillet, "Design of experiments for performance evaluation and parameter tuning of a road image processing chain," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 212–212, 2006.
21. J. Vogel and B. Schiele, "On performance characterization and optimization for image retrieval," *Proceedings of the European Conference on Computer Vision (ECCV2006)*, vol. 2353, pp. 51–55, 2006.
22. D. Burfoot, "Notes on a new philosophy of empirical science," *Arxiv preprint arXiv:1104.5466*, 2011.
23. A. Burton and J. Radford, *Thinking in perspective: critical essays in the study of thought processes.* Methuen, 1978.
24. H. Haussecker and H. Spies, "Motion," in *Handbook of Computer Vision and Applications* (B. Jähne, H. Haussecker, and P. Geissler, eds.), vol. 2, ch. 13, Academic Press, 1999.
25. D. Warren and E. Strelow, *Electronic spatial sensing for the blind: contributions from perception, rehabilitation, and computer vision.* No. 99, Kluwer Academic Print on Demand, 1985.
26. M. Raffel, C. Willert, and J. Kompenhans, "Postprocessing of PIV data," in *Particle Image Velocimetry*, ch. 6, Springer, 1998.
27. L. Maciaszek, *Requirements analysis and system design.* Pearson Education, 2007.
28. A. Kossiakoff, W. Sweet, S. Seymour, and S. Biemer, *Systems engineering principles and practice*, vol. 27. Wiley Online Library, 2003.
29. E. Mikhail, J. Bethel, and J. McGlone, *Introduction to modern photogrammetry*, vol. 31. Wiley New York, NY, 2001.
30. R. I. Hartley and A. Zisserman, *Multiple View Geometry.* Cambridge University Press, 2000.
31. T. Thormaehlen, "Zuverlässige schätzung der kamerabewegung aus einer bildfolge," 2006.
32. J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys, "Building rome on a cloudless day," in *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, (Berlin, Heidelberg), pp. 368–381, Springer-Verlag, 2010.
33. N. Guilherme and C. Avinash, "Vision for mobile robot navigation: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.
34. N. Ohnishi and A. Imiya, "Featureless robot navigation using optical flow," *Connection Science*, vol. 17, no. 1-2, pp. 23–46, 2005.
35. C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 600–608, ACM, 2004.
36. S. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 279–288, ACM, 1993.
37. P. Parsonage, A. Hilton, and J. Starck, "Efficient dense reconstruction from video," in *Proceedings of the 8th European Conference on Visual Media Production*, http://www.cvmp-conference.org/2011-Papers, 2011.

38. M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross, "Nonlinear disparity mapping for stereoscopic 3d," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 75, 2010.
39. C. Garbe and B. Jähne, "Reliable estimates of the sea surface heat flux from image sequences," *Pattern Recognition*, pp. 194–201, 2001.
40. J. Barron and A. Liptay, "Measuring 3-d plant growth using optical flow," *Bioimaging*, vol. 5, no. 2, pp. 82–86, 1997.
41. C. Kähler, B. Sammler, and J. Kompenhans, "Generation and control of tracer particles for optical flow investigations in air," *Experiments in fluids*, vol. 33, no. 6, pp. 736–742, 2002.
42. N. Papadakis, É. Mémin, *et al.*, "Variational assimilation of fluid motion from image sequence," *SIAM Journal on Imaging Science*, vol. 1, no. 4, pp. 343–363, 2008.
43. A. Berthe, D. Kondermann, C. Christensen, L. Goubergrits, C. Garbe, K. Affeld, and U. Kertzscher, "Three-dimensional, three-component wall-PIV," *Experiments in Fluids*, vol. 48, pp. 983–997, 2010.
44. C. Tropea, A. L. Yarin, and J. F. Foss, *Springer Handbook of Experimental Fluid Mechanics*. Springer, 2007.
45. A. Fincham and G. Spedding, "Low cost, high resolution dpiv for measurement of turbulent fluid flow," *Experiments in Fluids*, vol. 23, no. 6, pp. 449–462, 1997.
46. A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proc. of the 8th International Conference of Computer Vision, (ICCV03*, pp. 726–733, IEEE, 2003.
47. M. Haag and H. Nagel, "Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 295–319, 1999.
48. W. Wolf, "Key frame selection by motion analysis," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing, (ICASSP-96.*, vol. 2, pp. 1228–1231, Ieee, 1996.
49. G. Sudhir and J. Lee, "Video annotation by motion interpretation using optical flow streams," 1997.
50. A. Hauptmann, J. Gao, R. Yan, Y. Qi, J. Yang, and H. Wactlar, "Automated analysis of nursing home observations," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 15–21, 2004.
51. M. Michels, R. Rojas, and T. Landgraf, "A beehive monitoring system incorporating optical flow as a source of information," 2011.
52. B. Lombardot, M. Luengo-Oroz, C. Melani, E. Faure, A. Santos, N. Peyrieras, M. Ledesma-Carbayo, P. Bourgine, G. de Neurobiologie Alfred Fessard, and F. Yvette, "Evaluation of four 3d non rigid registration methods applied to early zebrafish development sequences," *MIAAB MICCAI*, 2008.
53. A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.
54. K. Krajsek and R. Mester, "Wiener-optimized discrete filters for differential motion estimation," in *Complex Motion. Proc. 1st International Workshop on Complex Motion, Günzburg, Germany, Oct. 2004* (B. Jähne, E. Barth, R. Mester, and H. Scharr, eds.), vol. 3417 of *Lecture Notes in Computer Science*, (Berlin), pp. 30–41, Springer Berlin Heidelberg, 2006.
55. R. M. Kai Krajsek and H. Scharr, "Statistically optimal averaging for image restoration and optical flow estimation," in *Pattern Recognition*, vol. 5096 of *Lecture Notes in Computer Science*, (Berlin / Heidelberg), pp. 466–475, Springer, 2008.

56. W. Forstner, “Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision,” *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 3, pp. 273–310, 1987.
57. S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, “Modeling temporal coherence for optical flow,” in *Proc. of the 13th International Conference of Computer Vision, (ICCV11)*, 2011.
58. F. Becker, F. Lenzen, J. H. Kappes, and C. Schnörr, “Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences,” in *Proc. of the 13th International Conference of Computer Vision, (ICCV11)*, 2011.
59. D. J. Fleet and A. Jepson, “Computation of component image velocity from local phase information,” *International Journal on Computer Vision*, vol. 5, no. 1, pp. 77–104, 1990.
60. B. Jähne, H. Haussecker, and P. e. Geißler, *Handbook of Computer Vision and Application. Volume 2.* Academic Press, 1999.
61. D. Heeger, “Model for the extraction of image flow,” *Journal of the Optical Society of America*, vol. 4, no. 8, pp. 1455–1471, 1987.
62. C. Kondermann, R. Mester, and C. Garbe, “A statistical confidence measure for optical flows,” in *Proc. of 10th Europian Conference of Computer Vision (ECCV 2008) Part III*, October.
63. A. Humayun, O. Mac Aodha, and G. Brostow, “Learning to find occlusion regions,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2161–2168, IEEE, 2011.
64. R. Bainbridge-Smith, A. Lane, “Measuring confidence in optical flow estimation,” *IET Electronics Letters*, vol. 32, no. 10, pp. 882–884, 1996.
65. C. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1995.
66. C. Zetzsche and E. Barth, “Fundamental limits of linear filters in the visual processing of two-dimensional signals,” *Vision Research*, vol. 30, no. 7, pp. 1111–1117, 1990.
67. S. Kalkan, D. Calow, M. Felsberg, F. Worgotter, M. Lappe, and N. Krüger, “Optic flow statistics and intrinsic dimensionality,” 2004.
68. M. Felsberg, S. Kalkan, and N. Krüger, “Continuous dimensionality characterization of image structures,” *Image and Vision Computing*, vol. 27, no. 6, pp. 628–636, 2009.
69. C. Kondermann, D. Kondermann, and C. Garbe, “Postprocessing of optical flows via surface measures and motion inpainting,” in *Pattern Recognition, (Proc. of DAGM2008)*, vol. 5096 of *LNCS*, pp. 355–365, Springer, 2008.
70. J. Kybic and C. Nieuwenhuis, “Bootstrap optical flow confidence and uncertainty measure,” *Computer Vision and Image Understanding*, vol. 115, no. 10, pp. 1449–1462, 2011.
71. M. Black, Y. Yacoob, A. Jepson, and D. Fleet, “Learning parameterized models of image motion,” in *Proc. of IEEE Computer Siciety Conference on Computer Vision and Pattern Recognition, (CVPR97)*, pp. 561–567, 1997.
72. S. Roth and M. Black, “On the spatial statistics of optical flow,” in *Proc. of International Conference on Computer Vision, (ICCV05)*, vol. 1, pp. 42–49, 2005.
73. D. Sun, S. Roth, J. P. Lewis, and M. J. Black, “Learning optical flow,” in *Proc. of the 10th European Conference on Computer Vision(ECCV08)*, (Berlin, Heidelberg), pp. 83–97, Springer-Verlag, 2008.

74. O. Mac Aodha, G. J. Brostow, and M. Pollefeys, "Segmenting video into classes of algorithm-suitability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR10)*, pp. 1054–1061, 2010.
75. S. Gehrig and T. Scharwächter, "A real-time multi-cue framework for determining optical flow confidence," in *Proc. of the 13th International Conference of Computer Vision, (ICCV11)*, 2011.
76. T. Amiaz, E. Lubetzky, and N. Kiryati, "Coarse to over-fine optical flow estimation," *Pattern Recogn.*, vol. 40, no. 9, 2007.
77. H. Scharr, "Optimal filters for extended optical flow," in *Complex Motion*, vol. 3417 of *LNCS*, Springer, 2004.
78. D. Kondermann, *Modular Optical Flow Estimation with Applications to Fluid Dynamics.* PhD thesis, University of Heidelberg, 2009.
79. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Artech House, 1999.
80. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr, *Real-Time Optic Flow Computation with Variational Methods*, pp. 222–229. Springer Berlin Heidelberg, 2003.
81. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr, "Real-time optic flow computation with variational methods," *IEEE Trans of Image Processing*, vol. 14, no. 5, pp. 608–615, 2005.
82. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic Huber-L1 optical flow," in *Proc. of the British Machine Vision Conference (BMVC09)*, (London, UK), September 2009.
83. S. Meister, B. Jähne, and D. Kondermann, "An outdoor stereo camera system for the generation of real-world benchmark datasets," *Optical Engineering (accepted)*, 2012.
84. T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," pp. 1–6, 2008.
85. C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, "Human-assisted motion annotation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR08)*, vol. 0, pp. 1–8, 2008.
86. B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, 2008.
87. S. Meister, "A study on ground truth generation for optical flow," Master's thesis, University of Heidelberg, 2010.
88. S. Meister and D. Kondermann, "Real versus realistically rendered scenes for optical flow evaluation.," in *Proceedings of 14th ITG Conference on Electronic Media Technology,*, 2011.
89. R. Haeusler and R. Klette, "Benchmarking stereo data (not the matching algorithms)," in *Pattern Recognition, (Proc. of DAGM10)* (M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler, eds.), vol. 6376 of *LNCS*, pp. 383–392, Springer Berlin-Heidelberg, 2010.
90. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR11)*, vol. 2, p. 3, 2011.
91. B. Kaneva, A. Torralba, and W. Freeman, "Evaluation of image features using a photorealistic virtual world," in *Proc. of the 13th International Conference of Computer Vision, (ICCV11)*, 2011.

92. B. Beizer, *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.