IJCV - Special Issue EMMCVPR manuscript No.
(will be inserted by the editor)

Stereoscopic Scene Flow Computation for 3D Motion Understanding

Andreas Wedel · Thomas Brox · Tobi Vaudrey · Clemens Rabe · Uwe Franke · Daniel Cremers

Received: date / Accepted: date

Abstract Building upon recent developments in optical flow and stereo matching estimation, we propose a variational framework for the estimation of stereoscopic scene flow, i.e., the motion of points in the three-dimensional world from stereo image sequences. The proposed algorithm takes into account image pairs from two consecutive times and computes both depth and a 3D motion vector associated with each point in the image. In contrast to previous works, we partially decouple the depth estimation from the motion estimation, which has many practical advantages. The variational formulation is quite flexible and can handle both sparse or dense disparity maps. The proposed method is very efficient; with the depth map being computed on an FPGA, and the scene flow computed on the GPU, the proposed algorithm runs at frame rates of 15 frames per second on QVGA images (320×240 pixels). Furthermore, we present solutions to two important problems in scene flow estimation: violations of intensity consistency between input images, and the uncertainty measures for the scene flow result.

1 Introduction

One of the most important features to extract in image sequences from a dynamic environment is the motion of points within the scene. Humans perform this using a process called visual kinesthesia, which encompasses both the perception of movement of objects in the scene and the observer's own movement. Perceiving this using computer vision based methods proves to be difficult. Images from a single camera are not well constrained. Only the perceived two-dimensional motion can be estimated from sequential images, commonly referred to as optical flow. Up to estimation errors and some well-known ambiguities (aperture problem), the optical flow corresponds to the three-dimensional scene motion projected to the image plane. The motion in depth is lost by the projection.

There are ways to recover the depth information from calibrated monocular video in a static scene with a moving observer up to a similarity transform. However, this requires a translating motion of the observer. The process becomes even more complex when there

A. Wedel
University of Bonn, Germany
E-mail: wedel@cs.uni-bonn.de

In International Journal of Computer Vision, 35(1):29-51, 2011
http://www.springerlink.com/content/028362k33p1560u5/fulltext.pdf

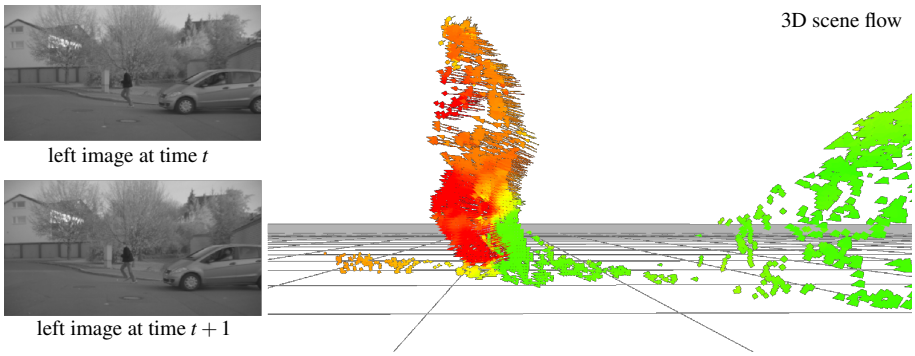


Fig. 1 The scene flow motion field depicted on the *right* is computed from two stereo input image pairs. On the *left*, the images from the left camera are shown for the two time instances. The colour encodes speed from stationary (green) to rapidly moving (red) after camera ego-motion compensation. Note the accurate motion estimation of the leg and the arm of the running person.

are independently moving objects in the scene. The estimation of the motion then has to be combined with a separation of independent motions - a chicken-and-egg problem which is susceptible to noise and/or local optima [9,20,28,44].

Once a stereo camera system is available, the task becomes better constrained and feasible in practice. The distance estimate from the triangulation of stereo correspondences provides vital additional information to reconstruct the three-dimensional scene motion. Then, ambiguities only arise

1. if the camera motion is not known, in particular the camera is not stationary. In this case, only the motion relative to the camera can be estimated.
2. when points are occluded
3. around areas with missing structure in a local neighbourhood

The first two ambiguities are quite natural and affect also human perception. For instance, human perception cannot distinguish whether the earth is rotating or the stars circulate around us[1]. The third ambiguity is well-known in both disparity estimation and optical flow estimation. A common way to deal with the missing structure and to achieve dense estimates is the use of variational approaches that incorporate a smoothness prior that resolves the ambiguity. In this sense, when we speak of *dense* estimates we mean that for each 3D point that is seen in both cameras, we have an estimate of its motion. Figure 1 is an example, where the movement of a running person becomes visible in the 3D scene flow.

Related Work

2D motion vectors are usually obtained by optical flow estimation techniques. Sparse techniques, such as KLT tracking [34], perform some kind of feature tracking and are preferred in time-critical applications. Dense optical flow is mostly provided by variational models based on the method of Horn and Schunck [16]. Local variational optimisation is used to minimise an energy functional that assumes constant pixel intensities and a smooth flow field. The basic framework of Horn and Schunck has been improved over time to cope with

[1] a fact that has nurtured many religious discussions

Algorithm	# cameras	Dense approach (yes/no)	Close / far	Running time
"Joint motion ..." [25]	2	Yes	Close	?
"Decouple: image segmentation ..." [47]	2	Yes	Close	?
Three-dimensional scene flow [38]	17	Yes	Close	?
6D Vision [27]	2	No	Both	40 ms
"Dense motion ..." [19]	2	Yes	Very close	5 s
"Multi view reconstruction ..." [26]	30	Yes	Close	10 min
Huguet-Devernay [18]	2	Yes	Both	5 hours
"Disparity flow ..." [11]	3	Yes	Close	80 ms
Decoupled (this paper)	2	Yes	Both	70 ms

Table 1 Scene flow algorithms with their running times, density, and range tested.

discontinuities in the flow field, and to obtain robust solutions with the presence of outliers in image intensities [3,23]. Furthermore, larger displacements can be estimated thanks to image warping and non-linearised model equations [23,6]. Currently, variational techniques yield the most accurate optical flow in the literature [39,43,48]. For the current state-of-the-art we refer to the Middlebury benchmark on optical flow [30]. Real-time methods have been proposed in [8,46].

Scene flow computation involves an additional disparity estimation problem, as well as the task of estimating the change of disparity over time. The work in [25] introduced scene flow as a joint motion and disparity estimation method. The succeeding works in [18,24,47] presented energy minimisation frameworks including regularisation constraints to provide dense scene flow. Other dense scene flow algorithms have been presented in multiple camera set-ups [26,38]. However, these only allow for non-consistent flow fields in single image pairs.

None of the above approaches run in real-time (see Table 1), giving best performances in the scale of minutes. The work in [19] presents a probabilistic scene flow algorithm with computation times in the range of seconds, but yielding only discrete integer pixel-accurate (not sub-pixel) results. [12] presented a discrete *disparity flow* (i.e., scene flow) algorithm that ran in the range of 1-2 seconds on QVGA (320×240 pixel) images. Real-time sub-pixel accurate scene flow algorithms, such as the one presented in [27], provide only sparse results both for the disparity and the displacement estimates.

The only real-time scene flow algorithm presented in the literature so far is the disparity flow algorithm in [11], which is an extension of [12]. This method is a discrete, combinatorial method and requires, *a-priori*, the allowed range (and discretisation) of values. It runs on QVGA images at 12 Hz using a local stereo method and at 5 Hz for the dynamic programming method (running on a GPU). Note that this is for a range of 40 discrete values (the author chose the range ± 5 for the optical flow component and ± 1 for the disparity flow), which is rather limited in its application. This method uses a decoupled approach for solving the disparity and the scene flow separately, but provides a loose coupling by predicting the disparity map using the scene flow with an error validation step.

In contrast, the method we present here provides sub-pixel accurate scene flow, due to the variational nature of the implementation, for any sized flow vector (handles both large and small vectors easily), and close to real-time (5 Hz on a CPU, 15 Hz on a GPU) for QVGA images. Parts of this work have been presented in three preliminary conference papers [40–42].

Contributions

Combining disparity estimation and motion estimation into one framework has been the common approach for scene flow computation (e.g., [18,25,47]). In this paper, scene flow is presented as an alternative to the work from [18], which is based on [6]. The main contribution is that we propose decoupling of the motion estimation from the disparity estimation while maintaining the stereo constraints. In addition to the decoupling, we elaborate on the intensity consistency assumption (or Lambertian reflectance assumption) that is part of optical flow and most scene flow algorithms. It is known to cause errors when real-world lighting conditions have variable effects on the different input images. In scene flow computation, where we match four images, the effects are even greater than in optical flow estimation. We present a solution based on residual images [36]. Finally, we provide uncertainty measures for every pixel, and present their use for object segmentation.

Why decoupling is advantageous

The decoupling of depth (disparity) and motion (optical flow and disparity change) estimation might look unfavorable at a first glance, but it has two important advantages. Firstly, the challenges in motion estimation and disparity estimation are quite different. With disparity estimation, thanks to the epipolar constraint, only an ordered scalar field needs to be estimated. This enables the use of optimization methods that guarantee global optima, such as dynamic programming or graph-cuts, to establish point correspondences. Optical flow estimation, on the other hand, requires the estimation of a vector field without ordered labels. In this setting, global optimization in polynomial time is not available. Another important difference is that motion vectors tend to be smaller in magnitude than disparities. This is valid for most applications and can be assured for all applications by minimizing the time delay in between the images) . Thus sub-pixel accuracy as provided by variational methods is more important for motion estimation, whereas occlusion handling is more critical in disparity estimation.

> Splitting scene flow computation into the estimation sub-problems, *disparity* and *optical flow with disparity change*, allows one to choose the optimal technique for each task.

It is worth noting that, although we separate the disparity estimation problem from the motion estimation, the proposed method still involves a coupling of these two tasks in the final scene flow computation, as the optical flow is enforced to be consistent with the computed disparities.

Secondly, the two sub-problems can be solved more efficiently than the joint problem. We find that the decoupling strategy allows for real-time computation of high-quality scene flow on the GPU with a frame rate of 20 Hz on QVGA images assuming the disparity map is provided (or implemented in hardware). On the CPU, we achieve 5 Hz.

> The splitting approach to scene flow is about 500 times faster compared to recent techniques for joint scene flow computation. [2]

How does the decoupling affect the quality of the results? In the decoupling strategy we propose, the motion field estimation takes into account the estimated disparities, but

[2] The exception is [11], which is a loosely coupled approach.

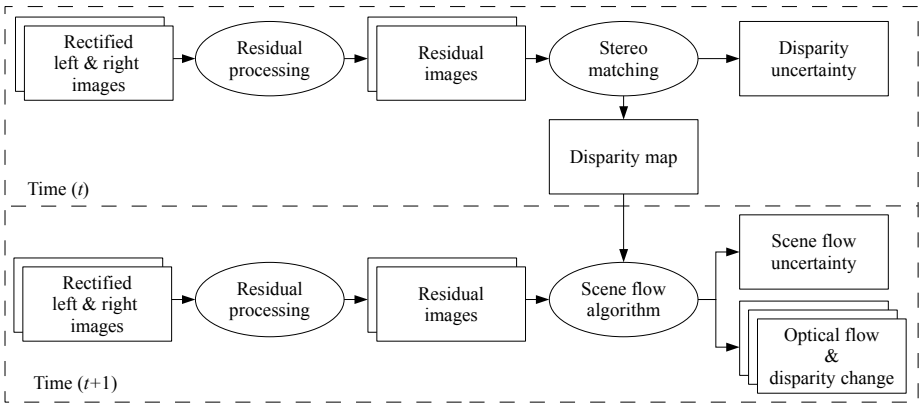


Fig. 2 Outline of our scene flow algorithm, showing the required data and information flow. The information needed to compute scene flow at frame t are the stereo pairs at t and $t+1$, along with the disparity map at t.

the disparity computation does not benefit from the computed motion fields. In coupled approaches like [18], all variables are optimized at the same time. However, variational optimization is a local approach, which is likely to run into local minima, especially when there are many coupled variables. Even though a coupled energy is an advantageous formulation of the problem, it is impossible to globally optimize this energy. In contrast, our disparity estimates are based on a global optimization technique. Although the disparities are not refined later in the variational approach anymore, they are more likely to be correct than in a coupled variational setting. This explains why the estimation accuracy of our decoupled approach actually turns out to compare favourably to that of joint estimation methods.

Paper organization

Section 2 first formulates the scene flow assumptions and energy equations, and then discusses implementation strategies, along with pseudo-code. In Section 3, we provide a way of estimating the pixel-wise uncertainty of our calculated results. This aims to be used for follow-on processes, such as filtering and segmentation. Section 4 takes the image based scene flow and formulates the transformation to real-world coordinates. Furthermore, this section provides metrics and likelihoods for understanding motion and speed. We present experimental results in Section 5 and conclude the paper in the final section.

2 Formulation of Scene Flow

Figure 2 shows the outline of the approach. As seen from this figure, disparity estimation is decoupled from the variational scene flow computation in order to allow for the use of efficient and optimal stereo algorithms. Given the disparity at t, we compute the optical flow and change in disparity from the two stereo pairs at time t and $t+1$. The disparity, optical flow and disparity change together determine scene flow.

2.1 Disparity Estimation

A disparity $d := d(x,y,t)$ is calculated for every pixel position $[x,y]^\top$ at every time frame t. Here, the pixel position is found in the image domain Ω. Current state-of-the-art algorithms (e.g., see Middlebury [30]) require normal stereo epipolar geometry, such that pixel row y for the left and right images coincide. This is achieved by a so-called rectification process given the fundamental matrix of the stereo camera [13].

A world point $[X,Y,Z]^\top$ (lateral, vertical and depth resp.) is projected into the cameras images, yielding $[x,y]^\top$ in the left image and $[x+d,y]^\top$ in the right image, according to:

$$\begin{pmatrix} x \\ y \\ d \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} Xf_x \\ -Yf_y \\ bf_x \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ 0 \end{pmatrix} \tag{1}$$

with the focal lengths f_x and f_y (in pixels) for the x and y direction, $[x_0,y_0]^\top$ is the principal point of the stereo camera system, and b is the baseline distance between the two camera projection centres (in metres). The disparity value d therefore encodes the difference in the x-coordinate of an image correspondence between the left and right image. With known intrinsic camera parameters (from calibration), the position of a world point $[X,Y,Z]^\top$ can be recovered from an $[x,y,d]^\top$ measurement using Eq. 1.

The goal of the stereo correspondence algorithm is to estimate the disparity d, for every non-occluded pixel in the left image. This is accomplished by local methods, which use a small matching window from the left to the right image, or global methods, which incorporate some global regularity constraints. The scene flow algorithm that we present later has the flexibility to use any disparity map as input. Dense or sparse algorithms are handled effectively due to the variational nature of the approach. In Section 5 we show scene flow results for different disparity estimation algorithms. One is a hierarchical correlation algorithm yielding sparse sub-pixel accurate disparity maps, which runs at about 100 Hz [10]. The other produces a sparse, pixel-discrete disparity map using Census based hash tables [32]. It is massively parallel and available in hardware (FPGA - field programmable gate array) without extra computational cost. Finally, we consider semi-global matching (SGM) with mutual information [15], a globally consistent energy minimisation technique that provides a disparity estimate for every non-occluded pixel. This algorithm is implemented on dedicated hardware (FPGA) and runs at 30 Hz on images with a resolution of 640×480 pixels. SGM is used as our main stereo algorithm.

2.2 Stereo Motion Constraints

The data dependencies exploited in the scene flow algorithm are shown in Figure 3. We use two consecutive pairs of stereo images at time t and $t+1$. The scene flow field $[u,v,p]^\top$ is an extension of the optical flow field $[u(x,y,t),v(x,y,t)]^\top$ (flow in the x and y direction respectively) by an additional component $p(x,y,t)$ that constitutes the disparity change.

Three-dimensional scene flow can be reconstructed for points, where both the image positions described by $[x,y,d]^\top$ and their temporal change described by $[u,v,p]^\top$ are known. d is estimated using an arbitrary stereo algorithm, see Section 2.1. The disparity change and the two-dimensional optical flow field have to be estimated from the stereo image pairs.

For all the equations derived for scene flow, we employ the normal optical flow *intensity consistency assumption*, i.e., the intensity should be the same in both images for the same

world point in the scene. We expand this to couple the four images involved with the scene flow calculation.

The first equation that we derive is from the left half of Figure 3. Let $L(x,y,t)$ be the intensity value of the left image, at pixel position $[x,y]^\top$ and time t. This leads to the following constraint, which we call the *left flow constraint*:

$$L(x,y,t) = L(x+u(x,y,t), y+v(x,y,t), t+1) \tag{2}$$

The flow in the right hand image can also be derived using the same principle. Let $R(x,y,t)$ be the intensity of the right image, at pixel position $[x,y]^\top$ and time t. Due to rectification, we know that flow in the left image and right image will have the same y component, this means that the difference is only in the x component of the equation. This leads to the *right flow constraint*:

$$R(x+d(x,y,t),y,t) = R(x+u(x,y,t)+d(x,y,t)+p(x,y,t), y+v(x,y,t), t+1) \tag{3}$$

highlighting that the position in the x component is offset by the disparity d and the flow is only different by the disparity change p.

Calculating optical flow in the left and right image separately, we could directly derive the disparity change $p = u^R - u^L$, where u^R and u^L denote the estimated flow fields in the left and right image, respectively. However, to estimate the disparity change more accurately, consistency of the left and right image at time $t+1$ is enforced. More precisely, the gray values of corresponding pixels in the stereo image pair at time $t+1$ should be equal, as illustrated in the bottom halve of the diagram in Figure 3. This yields the third constraint, the *disparity flow constraint*:

$$\begin{aligned} L(x+u(x,y,t), y+v(x,y,t), t+1) = \\ R(x+u(x,y,t)+d(x,y,t)+p(x,y,t), y+v(x,y,t), t+1) \end{aligned} \tag{4}$$

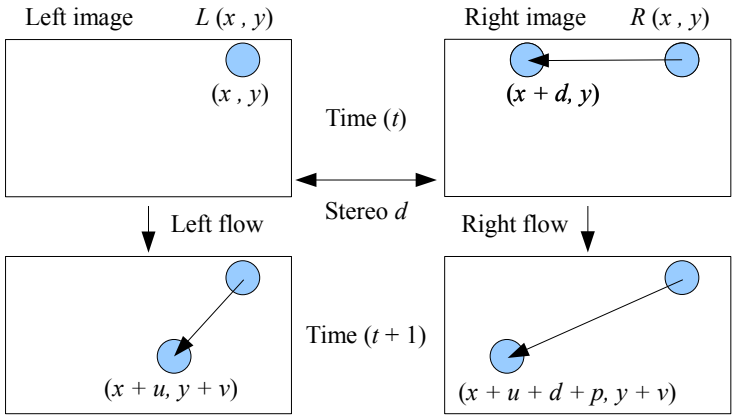


Fig. 3 Motion and disparity constraints employed in our scene flow algorithm. Intensity consistency of corresponding points in the left and right stereo image and in successive frames is assumed. This assumption is relaxed later in Section 2.5.

If we rearrange the above equations, it results in:

$$\begin{aligned} E_{LF} &:= L(x+u, y+v, t+1) - L(x,y,t) = 0 \\ E_{RF} &:= R(x+d+p+u, y+v, t+1) - R(x+d,y,t) = 0 \\ E_{DF} &:= R(x+d+p+u, y+v, t+1) - L(x+u, y+v, t+1) = 0 \end{aligned} \tag{5}$$

where the implicit dependency on (x,y,t) for u,v,p and d has been omitted. Figure 4 shows an illustration of the above equations in two real-world stereo pairs.

2.3 Energy Equations for Scene Flow

Scene flow estimates according to the constraints formulated in Section 2 are computed in a variational framework by minimising an energy functional consisting of a data term derived from the constraints and a smoothness term that enforces smoothness in the flow field; allowing for dense estimates in the image domain Ω despite sparse constraints:

$$E(u,v,p) = \int_{\Omega} \Big(E_{\mathrm{D}}(u,v,p) + E_{\mathrm{S}}(u,v,p) \Big) \mathrm{d}x\,\mathrm{d}y \tag{6}$$

By using the constraints from (5) we obtain the following data term:

$$E_{\mathrm{D}} = \Psi\left(E_{LF}^2\right) + c(x,y,t)\,\Psi\left(E_{RF}^2\right) + c(x,y,t)\,\Psi\left(E_{DF}^2\right) \tag{7}$$

where $\Psi\left(s^2\right) = \sqrt{s^2+\varepsilon^2}$ denotes the L^2 approximation for total variation L^1 that compensates for outliers [6] (with $\varepsilon = 0.01$ for numerical stability), and the function $c(x,y,t)$ returns 0 if there is no disparity given at $[x,y]^\top$, and 1 otherwise. This function deals with disparity maps, where data is missing, either due to a sparse stereo method or due to occlusion. Whenever $c(x,y,t) = 0$ the smoothness term defines the estimate at $[x,y,t]^\top$ taking into account the estimates of spatially neighbouring points. Due to this fill-in effect, the formulation provides dense image flow estimates $[u,v,p]^\top$, even if the disparity d is not dense.

The smoothness term penalises local deviations in the scene flow components and employs the same robust function as the data term in order to deal with discontinuities in the scene flow field:

$$E_{\mathrm{S}} = \lambda\,\Psi\left(|\nabla u|^2 + |\nabla v|^2\right) + \gamma\Psi\left(|\nabla p|^2\right) \quad \text{where} \quad \nabla := \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)^\top \tag{8}$$

and the parameters λ and γ regulate the importance of the smoothness constraint, with different weights assigned to the optical flow and the disparity change, respectively.

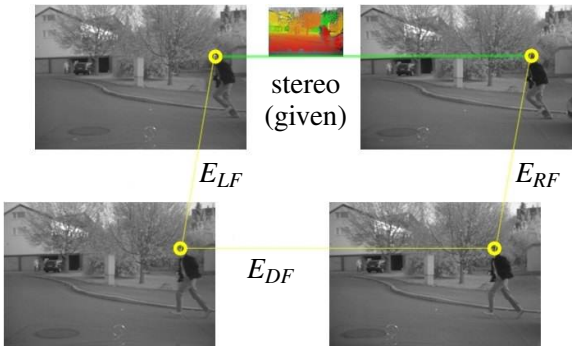


Fig. 4 The scene flow constraints (Eq. 5) are illustrated for two sequential stereo image pairs.

2.4 Minimisation of the Energy

For minimising the energy in the previous section we compute its Euler-Lagrange equations:

$$\Psi'\left(E_{LF}^2\right) E_{LF} L_x + c\,\Psi'\left(E_{RF}^2\right) E_{RF} R_x + c\,\Psi'\left(E_{DF}^2\right) E_{DF}\,(R_x - L_x) - \lambda\ \mathrm{div}\left(\nabla u\ E'_{\mathrm{S}}\right) = 0 \quad (9)$$

$$\Psi'\left(E_{LF}^2\right) E_{LF} L_y + c\,\Psi'\left(E_{RF}^2\right) E_{RF} R_y + c\,\Psi'\left(E_{DF}^2\right) E_{DF}\,(R_y - L_y) - \lambda\ \mathrm{div}\left(\nabla v\ E'_{\mathrm{S}}\right) = 0 \quad (10)$$

$$c\,\Psi'\left(E_{RF}^2\right) E_{RF} R_x + c\,\Psi'\left(E_{DF}^2\right) E_{DF} R_x - \gamma\ \mathrm{div}\left(\nabla p\ E'_{\mathrm{S}}\right) = 0 \quad (11)$$

with

$$E'_{\mathrm{S}} := \Psi'\left(\lambda\,|\nabla u|^2 + \lambda\,|\nabla v|^2 + \gamma|\nabla p|^2\right) \qquad \Psi'\left(s^2\right) = \frac{1}{\sqrt{s^2+\varepsilon^2}} \quad (12)$$

where $\Psi'\left(s^2\right)$ is the derivative of $\Psi\left(s^2\right)$ with respect to s^2. Partial derivatives of R and L are denoted by subscripts (see Algorithm 2 for details of calculation of R_* and L_*). Also the implicit dependency on (x,y,t) has been omitted from $c(x,y,t)$.

These equations are non-linear in the unknowns $[u,v,p]^\top$. We stick to the strategy of two nested fixed point iteration loops as suggested in [6]. The outer fixed point loop performs a linearisation of E_{LF}, E_{RF}, and E_{DF}. Starting with $[u^0,v^0,p^0]^\top = [0,0,0]^\top$, in each iteration k an increment $[\delta u^k, \delta v^k, \delta p^k]^\top$ of the unknowns is estimated and the second image is then warped according to the new estimate $[u^{k+1},v^{k+1},p^{k+1}]^\top = [u^k+\delta u^k, v^k+\delta v^k, p^k+\delta p^k]^\top$. The linearisation reads:

$$L(x+u^{k+1}, y+v^{k+1}, t) \approx L(x+u^k, y+v^k, t) + \delta u^k L_x + \delta v^k L_y \quad (13)$$

$$\begin{aligned} &R(x+d+p^{k+1}+u^{k+1}, y+v^{k+1}, t) \\ &\quad \approx R(x+d+p^k+u^k, y+v^k, t) + \delta u^k R_x + \delta p^k R_x + \delta v^k R_y \end{aligned} \quad (14)$$

From these expressions we can derive linearised versions of E_{LF}, E_{RF}, and E_{DF}. The warping is combined with a pyramid (coarse-to-fine) strategy, i.e., iterations start with downsampled versions of the image and the resolution is successively refined.

The remaining non-linearity in the Euler-Lagrange equations is due to the robust function. In the inner fixed point iteration loop the Ψ' expressions are kept constant and are recomputed after each iteration l. The resulting Euler-Lagrange equations together with implementation details on the implementation can be found in the Appendix, see Section 7.

2.5 Dealing with Intensity Consistency Assumption Violations

When dealing with synthetic scenes, the intensity consistency assumption (ICA) holds true. However, in "real-world" images, the ICA is usually not satisfied [37]. There have been several methods proposed to deal with this issue. [6] proposed using the intensity gradients in the data terms of the energy equations. Intensity gradients were shown to be approximately invariant to the most common intensity changes.

Alternative invariant features can be derived from a structure-texture image decomposition as introduced in [1]. The basic idea is to consider the residual image, which is the difference between the original image and a smoothed version of itself, thus removing low frequency illumination artifacts. The fact that this method works reasonably well for both

Fig. 5 An example of the residual images used as input to the scene flow algorithm in this paper. The original is on the left, with the residual processed image on the right.

stereo and optical flow [36] makes it the perfect synergy for scene flow. Any reasonable residual image can be used as shown in [36]. We use the TV-L^2 residual images for all real-world images (for synthetic data we use the original images as the ICA holds true), which are normalized to the range $L, R \in [-1, 1]$. This is a computationally expensive smoothing filter, but provides the most consistent results. Full details of the TV-L^2 residual image implementation is found in [29]. An example of a residual processed image is shown in Figure 5.

Comparing the residual image approach to adding gradient constancy constraints as in [6], residual images are favourable with regard to computational speed. They induce a smaller number of constraint equations in the energy and thus shorter Euler-Lagrange equations.

3 Uncertainty of Scene Flow

Understanding the uncertainty of an estimate is important for follow on processes, such as filtering and Markov-random field type segmentation. The uncertainty can be represented by variances. In the following subsections, we derive the uncertainty for the disparity and scene flow estimates for every pixel, by using the corresponding underlying energy functional.

3.1 Disparity Uncertainty

The scene flow algorithm presented above (primarily) uses the semi-global matching algorithm [14] for the disparity estimation and a variational framework for the scene flow estimates. The core semi-global matching algorithm produces discrete pixel-accurate results, with a sub-pixel interpolation performed afterward.

Let i be the disparity estimate of the core SGM (or any pixel discrete algorithm using energy minimization) method for a certain pixel in the left image. The SGM method in [14] is formulated as an energy minimization problem. Hence, changing the disparity by ± 1 yields an increase in costs (yielding an increased energy). The minimum, however, may be located in between pixels, motivating a subsequent sub-pixel estimation step. Sub-pixel accuracy is achieved by a subsequent fit of a symmetric equiangular function (see [31]) in the cost volume. Note, that this is different to standard linear interpolation of the disparity values as the fit is done in the cost volume and is aimed at finding the minimum of a symmetric $L1$ cost function approximating the costs for the three disparity assumptions i-1, i, and i+1.

The basic idea of this step is illustrated in Figure 6 for an example of a typical d estimate. This fit is unique and yields a specific sub-pixel minimum, located at the minimum of the function. Note that this might not be the exact minimum of the underlying energy but is a close approximation, evaluating the energy only at pixel position and assuming that the underlying energy function is smooth.

The slope of this fitting function (the larger of the two relative cost differences between the current estimate and neighbouring costs, Δj) serves as a quality measure for the goodness-of-fit. If the slope is low, the disparity estimate is not accurate in the sense that other disparity values could also be valid. If on the other hand the slope is large, the sub-pixel position of the disparity is expected to be quite accurate as deviation from this position increases the energy. Hence, the larger the slope, the better is the expected quality of the disparity estimate.

Based on this observation an uncertainty measure is derived for the expected variance of the disparity estimate:

$$U_D(x,y,d) = \frac{1}{\Delta j} \tag{15}$$

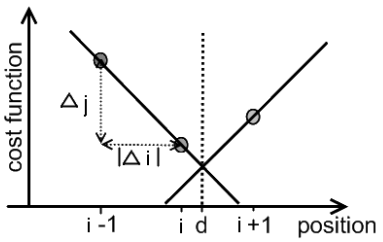


Fig. 6 Quality measure for the disparity estimate. The slope of the disparity cost function serves as a quality measure for the disparity estimate.

The disparity uncertainty could be calculated using the underlying energy equations, as is done for the scene flow below. However, the method using the slope of the interpolated function (presented above) has been shown to provide better results than when using the energy cost (outlined below). Other stereo uncertainty measures can be used, such as those presented in [17], and may provide better results with additional computational cost.

3.2 Scene Flow Uncertainty

For variational optical flow methods the idea of using the incline of the cost function or energy function as uncertainty measure becomes more complex than in the disparity setting. This is due to the higher dimensionality of the input and solution space. An alternative, energy-based confidence measure was proposed in [7]. The novel idea is that the uncertainty is proportional to the local energy contribution in the energy functional, used to compute the optical flow. A large contribution to the total energy implies high uncertainty (thus low accuracy), while uncertainty is expected low if the energy contribution is small. The authors show that this energy-based measure yields a better approximation of the *optimal* confidence for optical flow estimates than an image-gradient-based measure.

The same idea is now applied to the scene flow case. The three data terms in the energy functional are the *left*, *right*, and *disparity flow* constraints. Additionally, the smoothness of the scene flow variables u, v, and p also contribute to the energy functional, thus the uncertainty. This yields an expected uncertainty of the scene flow estimate:

$$U_{SF}(x,y,d,u,v,p) = E_{LF} + E_{RF} + E_{DF} + E_S \tag{16}$$

The main advantage of this uncertainty measure is that it is provided with out any additional computation. The cost (including both the intensity data and smoothness term) at each pixel is used in the algorithm, and the final value is used as the final uncertainty measure.

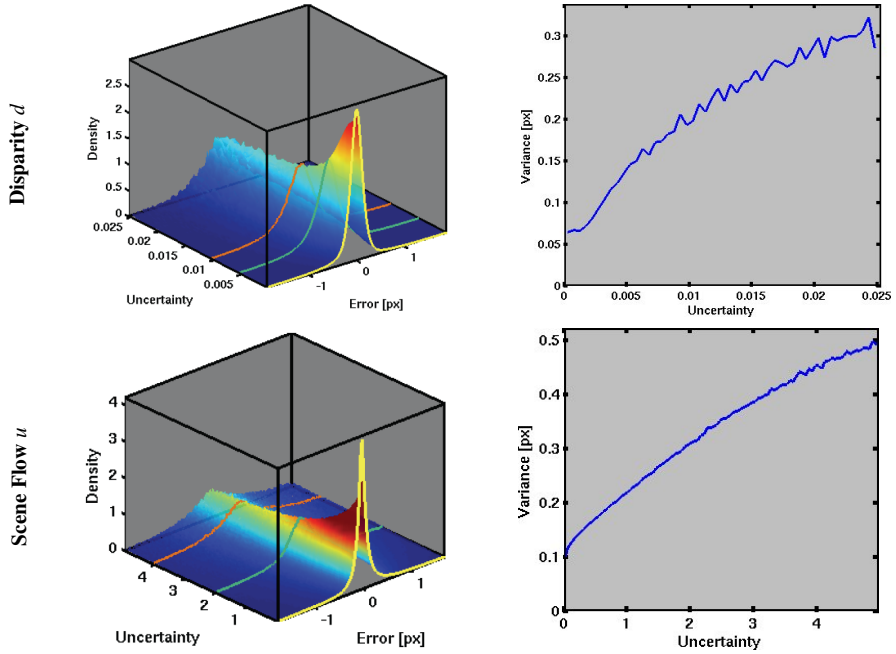


Fig. 7 Plots of the proposed uncertainty measures and corresponding variances for the disparity. The left column shows the density of actual error (using ground truth) vs. the uncertainty measure. The distribution at each uncertainty level is normalised to 1. The right column shows the true variance vs. the uncertainty measure (basically a summary of the density graph). VAR(d) vs. U_D, is shown at the top. The scene flow u-component VAR(u) vs. U_{SF} is shown at the bottom.

3.3 Comparing Variances and Uncertainty Measures

To evaluate the uncertainty measures for the disparity and scene flow estimates, we plot the derived uncertainty measures against the observed error in Figure 7 (for the disparity d and the u-component of the optical flow). To generate the plots we used a 400 frame evaluation sequence (sequence 2 from set 2 on [35]), which is outlined in Section 5.3.

The plots illustrate that the proposed uncertainty measures are correlated to the true variance of the errors. Furthermore, the variance σ_α for a scene flow component (where $\alpha \in \{d,u,v,p\}$) can be approximated by a linear function of the uncertainty measure, denoted by U_α, with fixed parameters g_α and h_α: $\sigma_\alpha^2(x,y,t) = g_\alpha + h_\alpha U_\alpha(x,y,t)$.

4 From Image Scene Flow to 3D Scene Flow (World Flow)

This section proposes methods for evaluating our scene flow algorithm. This involves first taking our scene flow (image coordinates) estimate, then estimating three-dimensional scene flow / world flow (real-world coordinates). We also propose two metrics for estimating confidence of moving points within the scene.

We have now derived the image scene flow as a combined estimation of optical flow and disparity change. Using this information, we can compute two world points that define the

start and end point of the 3D scene flow. These equations are derived from the inverse of Eq. 1 (f_x,f_y, and b are defined there as well).

$$X_t = (x - x_0)\frac{b}{d}, \quad Y_t = -(y - y_0)\frac{f_y}{f_x}\frac{b}{d}, \quad Z_t = \frac{f_x b}{d} \tag{17}$$

and

$$X_{t+1} = (x + u - x_0)\frac{b}{d+p}, \quad Y_{t+1} = -(y + v - y_0)\frac{f_y}{f_x}\frac{b}{d+p}, \quad Z_{t+1} = \frac{f_x b}{d+p} \tag{18}$$

Obviously, the 3D scene flow $[\dot{X},\dot{Y},\dot{Z}]^\top$ is the difference between these two world points. For simplicity we will assume that $f_y = f_x$. This yields:

$$\begin{pmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{pmatrix} = \begin{pmatrix}X_{t+1} - X_t\\ Y_{t+1} - Y_t\\ Z_{t+1} - Z_t\end{pmatrix} = b\begin{pmatrix}\frac{x+u-x_0}{d+p} - \frac{x-x_0}{d}\\ \frac{y-y_0}{d} - \frac{y+v-y_0}{d+p}\\ \frac{f_x}{d+p} - \frac{f_x}{d}\end{pmatrix} \tag{19}$$

For follow-on calculations (e.g., speed, detection of moving objects, segmentation of objects, integration, etc.) we need the accuracy of the scene flow vector. In the following subsections, we will define two metrics that estimate the likelihood that a pixel is moving, i.e., not stationary. The metrics provided in this section are ideas for follow-on evaluations, such as segmentation and filtering. An example of segmentation is shown in Section 5.5.

4.1 Residual Motion Likelihood

First, we define the uncertainty of our measurements. These can be represented by standard deviations as σ_d, σ_u, σ_v, and σ_p (subscript denoting variable of standard deviation). Discussion and results of what values to use for σ_α (where $\alpha \in \{d,u,v,p\}$) are found in Section 5.5. One would assume that $\sigma_u = \sigma_v \propto \sigma_p$ and the disparity estimate to be less accurate than the flow. We do not explicitly assume this and derive the covariance matrix Σ_{SF} for the 3D scene flow using error propagation:

$$\Sigma_{SF} = \mathbf{J}\ \mathrm{diag}(\sigma_d^2, \sigma_u^2, \sigma_v^2, \sigma_p^2)\ \mathbf{J}^\top \tag{20}$$

where

$$\mathbf{J} = \begin{pmatrix}\frac{\partial\dot{X}}{\partial d} & \frac{\partial\dot{X}}{\partial u} & \frac{\partial\dot{X}}{\partial v} & \frac{\partial\dot{X}}{\partial p}\\ \frac{\partial\dot{Y}}{\partial d} & \frac{\partial\dot{Y}}{\partial u} & \frac{\partial\dot{Y}}{\partial v} & \frac{\partial\dot{Y}}{\partial p}\\ \frac{\partial\dot{Z}}{\partial d} & \frac{\partial\dot{Z}}{\partial u} & \frac{\partial\dot{Z}}{\partial v} & \frac{\partial\dot{Z}}{\partial p}\end{pmatrix} = \mathbf{b}\begin{pmatrix}\left(\frac{(x+u-x_0)}{(d+p)^2} - \frac{(x-x_0)}{d^2}\right) & \frac{-1}{d+p} & 0 & \frac{(x+u-x_0)}{(d+p)^2}\\ \left(\frac{(y+v-y_0)}{(d+p)^2} - \frac{(y-y_0)}{d^2}\right) & 0 & \frac{-1}{d+p} & \frac{(y+v-y_0)}{(d+p)^2}\\ \left(\frac{f_x}{(d+p)^2} - \frac{f_x}{d^2}\right) & 0 & 0 & \frac{f_x}{(d+p)^2}\end{pmatrix} \tag{21}$$

This error propagation holds true, as long as the distribution is zero-mean and scales by the standard deviation (e.g., Gaussian and Laplacian). We have also assumed that the covariances are negligible, although the problem of scene flow estimation is highly coupled. However, estimating covariances is not trivial and in our eyes even impossible. We agree that errors in the flow x-component tends to impose errors on the y-component and disparity change but we do not know of a way to determine whether this correlation is positive or negative. From this model, one can see that the disparity measurement has the highest

influence on the covariance (as it is either by itself or quadratically weighted in the equations). Furthermore, the larger the disparity, the more precise the measurement; as $d \to \infty$ all $\sigma_\alpha \to 0$.

The derivation above only holds true for stationary cameras. Assume the motion of the camera (in our case a vehicle) is given from either inertial sensors or an ego-motion estimation method (e.g., [2]). This motion is composed of a rotation $\mathbf{R}$ (matrix composed by the combination of rotations about the X, Y and Z axis) about the origin of the camera coordinate system and a translation $\mathbf{T} = [T_X, T_Y, T_Z]^\top$. The total residual motion vector $\mathbf{M}$ is calculated as:

$$\mathbf{M} = \begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix} = \begin{pmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{pmatrix} - \mathbf{R} \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} + \mathbf{T} \tag{22}$$

Again the motion is known with a certain accuracy. For simplicity we assume the rotational parts to be small, which holds true for most vehicle applications. This approximates the rotation matrix by a unary matrix for the error propagation calculation. We denote the standard deviations of the translations as a three-dimensional covariance matrix Σ_T. The total translation vector vector $\mathbf{M}$ now has the covariance matrix $\Sigma_M = \Sigma_{SF} + \Sigma_T$.

Now one can compute the likelihood of a flow vector to be moving, hence belonging to a moving object. Assuming a stationary world and a Gaussian error propagation, one expects a standard normal distribution with mean 0 and covariance matrix Σ_M. Deviations from this assumption are found by testing this null hypothesis or the goodness of fit. This can be done by evaluating the Mahalanobis distance [22], giving us the *residual motion likelihood*:

$$\xi_M(x,y) = \sqrt{\mathbf{M}^\top \Sigma_M^{-1} \mathbf{M}} \tag{23}$$

The squared Mahalanobis distance ξ_M is χ^2 distributed and outliers are found by thresholding, using the assumed quantiles of the χ^2 distribution. For example, the 95% quantile of a distribution with three degrees of freedom is 7.81, the 99% quantile lies at 11.34. Hence a point is moving with a probability of 99% if the Mahalanobis distance is above 11.34. This again holds only if the measurement variances are correct. Figure 8 demonstrates results using this metric. In both images, it is easy to identify what parts of the scene are static, and which parts are moving. The movement metric ξ_M only identifies the probability of a point being stationary, it does not provide any speed estimates. Note that this metric computes a value at every scene point. Another two examples of results obtained using this metric can be seen in Figure 9.

4.2 Speed Metrics

The residual motion likelihood metric ξ_M omitted any information about speed. To estimate the speed S the L^2-norm (length) of the displacement vector is calculated.

$$S = \|\mathbf{M}\| \tag{24}$$

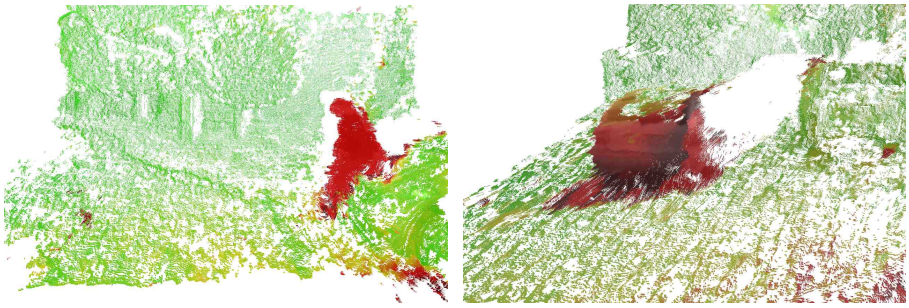
The problem is that points at large distances are always estimated as moving. This is because a small disparity change yields large displacements in 3D (see Equation 19). If inaccuracies are used in the residual motion computation one can still not derive speed information. One way around the problem is to give a lenient variance of the speed measurement

σ_S^2. An approach to estimate this variance is to calculate the *spectral norm* of the covariance matrix. This involves computing the eigenvalues of the squared matrix, then taking the square root of the maximum eigenvalue.

$$\sigma_S^2 = \|\Sigma_M\| = \sqrt{\lambda_{\max}\left(\Sigma_M^\top \Sigma_M\right)} \tag{25}$$

Using this we now have a likely speed S and associated variance σ_S^2. Using these metrics leads to the examples in Figure 10. In this figure, it is easy to identify the speed of moving targets, and also how confident we are of the speed measurement. The pedestrian in Figure 10(a) had a displacement of 15 cm with a frame rate of 25 Hz, i.e., 3.75 m/s. The vehicle in 10(b) had a displacement of 50cm, i.e., 12.5 m/s. In both examples only the information with high confidence is taken into account, so moving objects are easily identified.

From the metrics provided in this section, we now have a likelihood that the object is moving ξ_M, likely speed of the object S and the uncertainty of the speed σ_S^2.



(a) Moving Pedestrian (b) Lead Vehicle

Fig. 8 Results using the Manhalobis distance likelihood ξ_M. (a) shows a pedestrian running from behind a vehicle. (b) shows a lead vehicle driving forward. Colour encoding is ξ_M, i.e., the hypothesis that the point is moving, green $\leftrightarrow$ red $\equiv$ low $\leftrightarrow$ high.

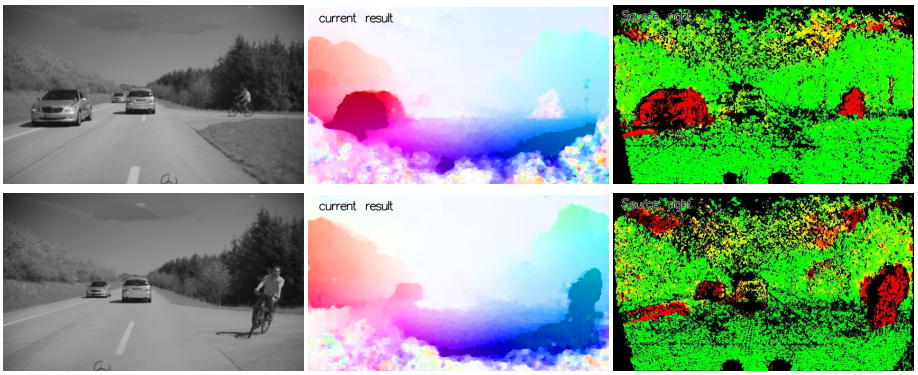
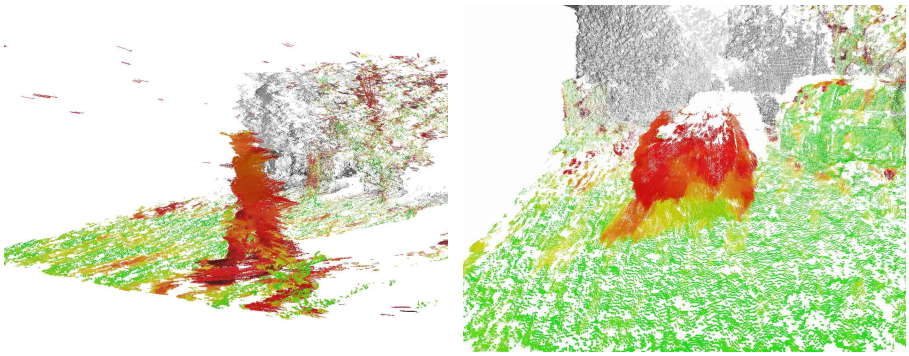


Fig. 9 Two examples, from a single sequence, of the residual motion likelihood defined in Section 4.1. Left to right: original image, optical flow result, the residual motion metric results (green $\leftrightarrow$ red represents low $\leftrightarrow$ high likelihood that point is moving).



(a) Moving Pedestrian (b) Lead Vehicle

Fig. 10 Results using speed S and its standard deviation σ_S. (a) shows the pedestrian running with a speed of 3.75 m/s. (b) shows a lead vehicle driving forward with a speed of 12.5 m/s. Colour encoding is S, green $\leftrightarrow$ red $\equiv$ stationary $\leftrightarrow$ moving. σ_S is encoded using saturation, points in the distance are therefore grey or black.

5 Evaluation and Experimental Results

The evaluation metrics used in this section are defined as follows. First we define error metrics at each time frame. The absolute angular error (AAE) as used in [18]:

$$AAE_{u,v} = \frac{1}{|\Omega|}\sum_{\Omega} \arctan\left(\frac{u\tilde{v} - \tilde{u}v}{u\tilde{u} + v\tilde{v}}\right) \tag{26}$$

where an accent $\tilde{\alpha}$ denotes the ground truth solution of α (where $\alpha \in \{d,u,v,p\}$) and $|\Omega|$ is the cardinality (number of pixels) in Ω.

The root mean square (RMS) error:

$$RMS_{u,v,d,p} = \sqrt{\frac{1}{|\Omega|}\sum_{(x,y)\in\Omega} \|[u,v,d,p]^\top - [\tilde{u},\tilde{v},\tilde{d},\tilde{p}]^\top\|^2} \tag{27}$$

If there is no disparity measure d (either by sparse algorithm or occlusion) then the estimated value is set to 0 (therefore still contributing to error). In our notation for RMS, if a subscript is omitted, then both the respective ground truth and estimated value are set to zero.

The 3D angular error:

$$AAE_{3D} = \frac{1}{|\Omega|}\sum_{\Omega} \arccos\left(\frac{u\tilde{u} + v\tilde{v} + p\tilde{p} + 1}{\sqrt{(u^2+v^2+p^2+1)\,(\tilde{u}^2+\tilde{v}^2+\tilde{p}^2+1)}}\right) \tag{28}$$

When evaluating errors over a time period, we use the following mean and variance of the error:

$$\mu(\alpha) = \frac{1}{|\Omega|}\sum_{\Omega} |\alpha - \tilde{\alpha}| \tag{29}$$

$$\sigma^2(\alpha) = \frac{1}{|\Omega|}\left(\sum_{\Omega} (\alpha - \tilde{\alpha})^2\right) - (\mu(\alpha))^2 \tag{30}$$

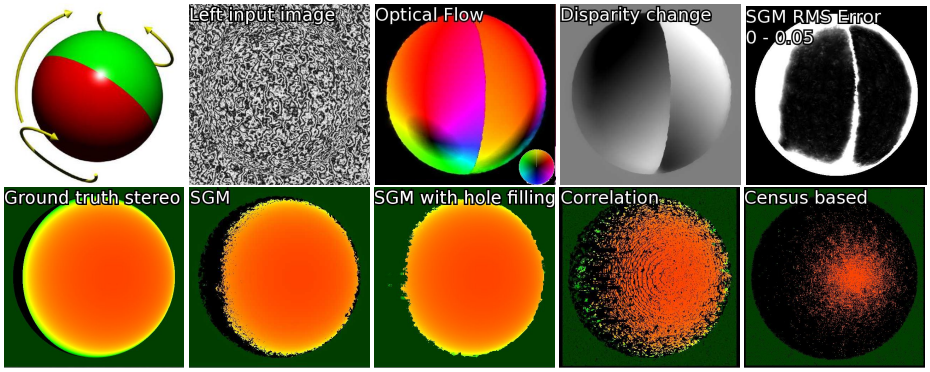


Fig. 11 Ground truth test: *rotating sphere*. Quantitative results are shown in Table 2. **Top:** The left image shows the movement of the sphere. Optical flow and disparity change are computed on the basis of SGM stereo [15]. Colour encodes the direction of the optical flow (key in bottom right), intensity its magnitude. Disparity change is encoded from black (increasing) to white (decreasing). Bright parts of the RMS figure indicate high $RMS_{u,v,p}$ error values of the computed scene flow. **Bottom:** disparity images are colour encoded green to orange (low to high). Black areas indicate missing disparity estimates or occluded areas.

Note, the implicit dependence on (x,y,t) for all $\alpha \in \{d,u,v,p\}$ is omitted from the above equations.

The first two subsections here present a summary of synthetic data testing, parts of these results have been published in [41]. The third subsection presents an evaluation approach for long stereo sequences, along with some sample results. Real-world results of the algorithm are provided in the fourth subsection. Finally, segmentation using residual motion likelihoods from Section 4.1 is presented.

All input images are 12-bit per pixel except the rotating sphere sequence. The finer quantization helps increasing the sub-pixel accuracy of the approach.

5.1 Rotating Sphere

To assess the quality of our scene flow algorithm, it was tested on synthetic sequences, where the ground truth is known. The first ground truth experiment is on the *rotating sphere* sequence from [18] depicted in Figure 11. In this sequence the spotty sphere rotates around its y-axis to the left, while the two hemispheres of the sphere rotate in opposing vertical directions[3]. The resolution is 512×512 pixels.

We tested the scene flow method together with four different stereo algorithms: semi-global matching (SGM [15]), SGM with hole filling (favours smaller disparities), correlation pyramid stereo [10], and an integer accurate census-based stereo algorithm [32]. The ground truth disparity was also used for comparison, i.e., using the ground truth as the input disparity for our algorithm.

The errors were calculated in two different ways: firstly, calculating statistics over all non-occluded areas, and secondly calculating over the whole sphere. As in [18], pixels from the background were not included in the statistics.

[3] The authors thank Huguet and Devernay for providing their *sphere scene*.

Stereo algorithm	RMS_d (density)	Without occluded areas			With occluded areas		
		$RMS_{u,v}$	$RMS_{u,v,p}$	$AAE_{u,v}$	$RMS_{u,v}$	$RMS_{u,v,p}$	$AAE_{u,v}$
Ground truth	0 (100%)	0.31	0.56	0.91	0.65	2.40	1.40
SGM [15]	2.9 (87%)	0.34	0.63	1.04	0.66	2.45	1.50
Correlation [10]	2.6 (43%)	0.33	0.73	1.02	0.65	2.50	1.52
Census based [32]	7.8 (16%)	0.32	1.14	1.01	0.65	2.68	1.43
Hug.-Dev. [18]	3.8 (100%)	0.37	0.83	1.24	0.69	2.51	1.75
Fill-SGM	10.9 (100%)	0.45	0.76	1.99	0.77	2.55	2.76

Table 2 Root mean square (pixels) and average angular error (degrees) for scene flow of the *rotating sphere* sequence. Various stereo algorithms are used as input for our scene flow estimation, generating varying results.

The resulting summary can be seen in Table 2. We achieve lower errors than the Huguet and Devernay method, even when using sparse correlation stereo. The lower error is even more *due* to the sparseness of the disparity since problematic regions such as occlusions are not included in the computation and can therefore not corrupt the estimates. Thanks to the variational formulation of the scene flow, more reliable information is filled in where the data terms are disables and a dense scene flow is obtained. Particularly, the RMS error of the scene flow is much smaller and we are still considerably faster (see Table 1). In this sense SGM seems to do a good job at avoiding occluded regions.

The joint approach in [18] is bound to the variational setting, which usually does not perform well for disparity estimation. Moreover the table shows that SGM with hole filling yields inferior results to the other stereo methods. This is due to false disparity measurements in the occluded area. It is better to feed the sparse measurements of SGM to the variational framework, which yields dense estimates as well, but with higher accuracy.

SGM was chosen as the best method and is used in the remainder of the results section; it is available on dedicated hardware without any extra computational cost.

5.2 Povray Traffic Scene 1

In a second ground truth example we use a Povray-rendered traffic scene [37], which is publicly available online for comparison [35]. The scene layout is shown in Figure 12. We calculated the $RMS_{u,v,p}$ error and the 3D angular error defined in Equation 28.

Results are shown in Figures 12 and 13. They compare favourably to the results obtained when running the code from [18]. The average $RMS_{u,v,p}$ error for the whole sequence (subregion as in Figure 12) was 0.64 pixels and the 3D angular error was 3.0°. The area of interest Ω is $x \in [50,590]$ and $y \in [50,400]$ with $\{x,y\} \in \mathbb{N}$ (images are $x \times y = 640 \times 480$px).

5.3 Evaluation Approach using Stereo Synthetic Data

In this subsection the Povray Traffic Scene 2 is used to analyse the output from our scene flow approach. It is a more complex driving scene, involving hills, trees, and realistic physics; it consists of 400 sequential stereo image pairs. An example picture is shown in Figure 14. This scene is publicly available with ground truth disparity, optical flow, disparity change, and ego-motion [35].

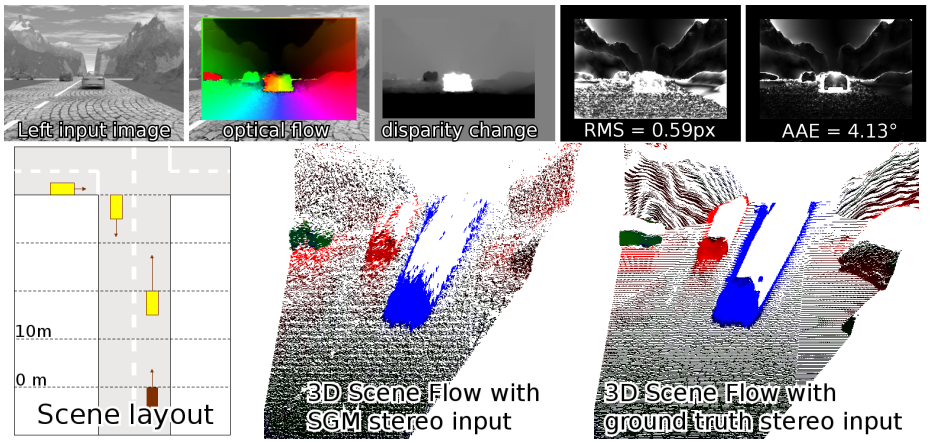


Fig. 12 Povray-rendered traffic scene (Frame 11). **Top:** Colour encodes direction (border = direction key) and intensity the magnitude of the optical flow vectors. Brighter areas in the error images denote larger errors. For comparison, running the code from [18] generates an RMS error of 0.91px and AAE of 6.83°. **Bottom right:** 3D views of the scene flow vectors. Colour encodes their direction and brightness their magnitude (black = stationary). The results from the scene are clipped at a distance of 100m. Accurate results are obtained even at greater distances.

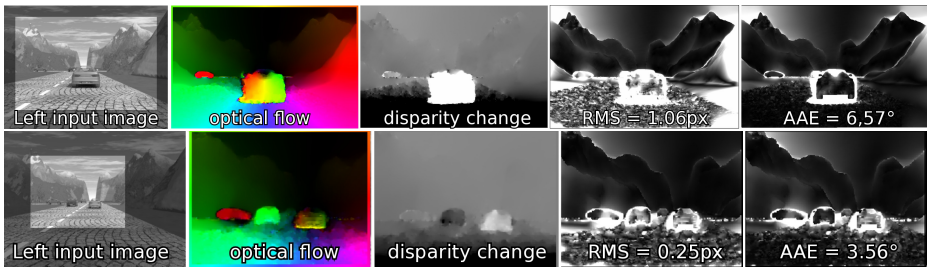


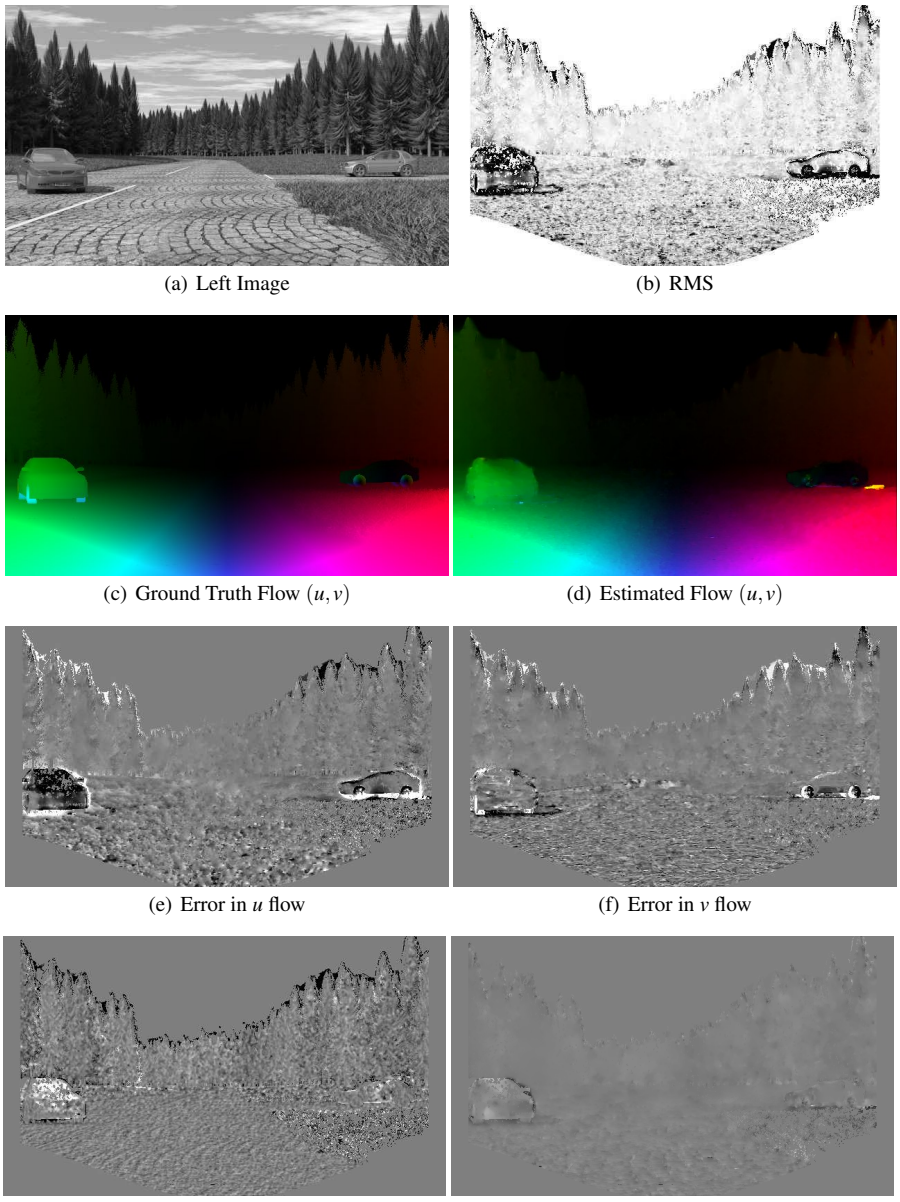
Fig. 13 More frames from the traffic scene in Figure 12. The top row highlights the problems such as transparency of the windshield, reflectance, and moving shadows. The bottom row demonstrates that we still maintain accuracy at distances of 50 m.

An evaluation approach has been devised to handle such a large dataset. For each image pair at time t, the following quantities are computed:

- $RMS_{u,v,p}$ from the subsection above.
- Error at each pixel, i.e., difference between estimate and ground truth.
- Absolute mean error for u,v,p, and d.
- Variance of the error for u,v,p, and d.

We have provided an example using our scene flow algorithm in Figure 14. From this figure it can be seen that the major errors are on object boundaries, and the errors in p are the lowest in magnitude.

The quantitative results for the entire image sequence are shown in the graphs of Figures 15 and 16. Comparing the errors of all four scene flow components in Figure 15, it can be seen that the errors for u and v are consistently about the same error as for d, except for a few frames (60, 210, 360, and 380) where their error is dramatically higher. The errors around frame 60, 360, and 380 all relate to the same issue; when previously occluded ar-



(a) Left Image (b) RMS

(c) Ground Truth Flow (u, v) (d) Estimated Flow (u, v)

(e) Error in u flow (f) Error in v flow

(g) Error in disparity d (h) Error in p flow

Fig. 14 Povray-rendered traffic scene 2 (Frame 215). RMS encoded low to high as light to dark brightness. Flow colour encoded as in Figure 11. Error images encoded with brightness, negative = dark, positive = light, zero = mid-grey value (e.g., infinite background). Points at infinity and occlusions in RMS and error images are shown as a zero value in each colour scale.

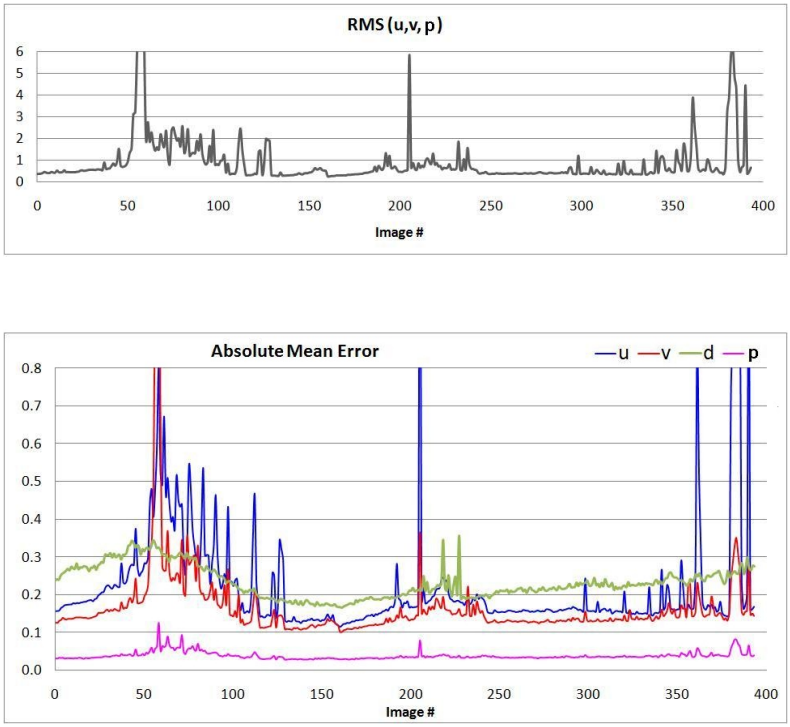
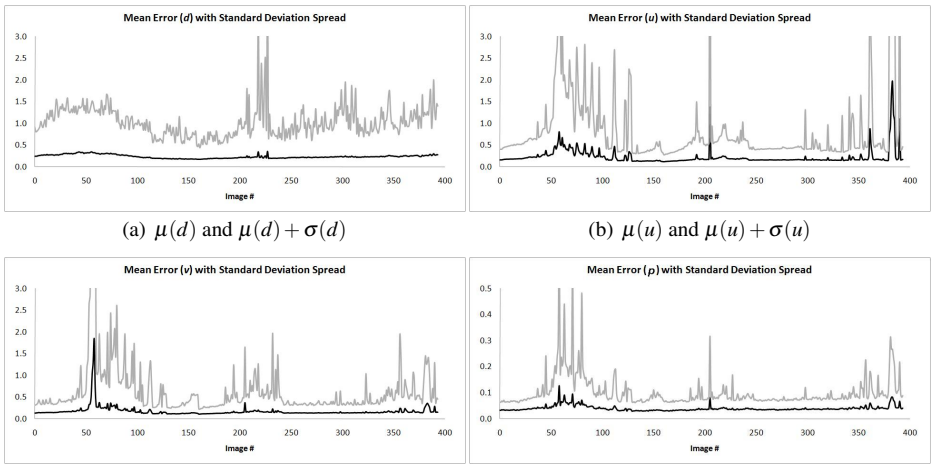


Fig. 15 RMS error and mean error evaluation over entire sequence.



(a) $\mu(d)$ and $\mu(d)+\sigma(d)$ (b) $\mu(u)$ and $\mu(u)+\sigma(u)$

(c) $\mu(v)$ and $\mu(v)+\sigma(v)$ (d) $\mu(p)$ and $\mu(p)+\sigma(p)$

Fig. 16 Graphs using the mean error and variance from Section 5.3. The graphs shows the results for disparity d and the flow estimates u,v, and p. The mean error (e.g., $\mu(d)$) is the dark coloured line. The light line is 1 standard deviation from the mean (e.g., $\mu(d)+\sigma(d)$).

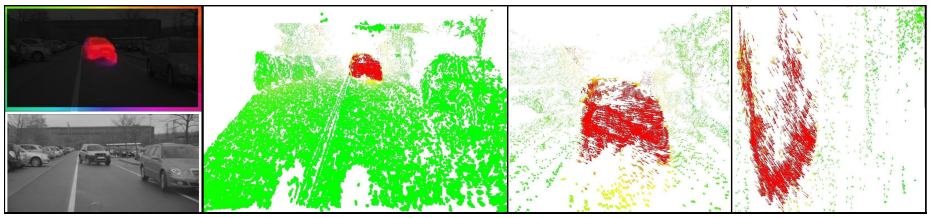


Fig. 17 This figure shows the scene flow results when on a stationary platform. From left to right: optical flow (top) and original image (bottom), scene flow 3D vectors, zoomed in 3D flow vectors, zoomed in 3D flow vectors when viewed from above. 3D vectors are coloured green ↔ red as stationary ↔ moving

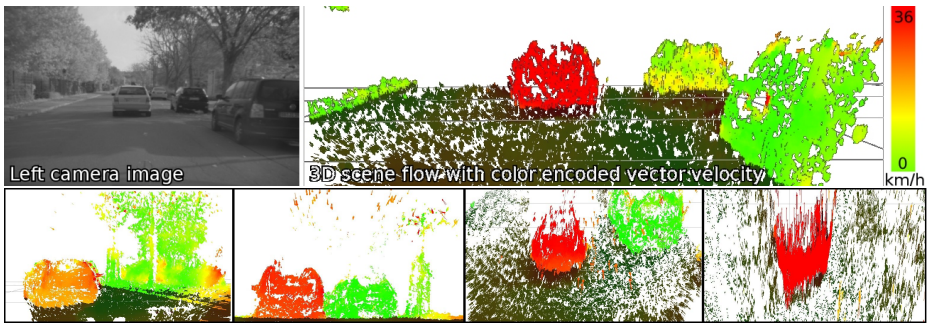


Fig. 18 This figure shows the scene flow results when following another vehicle. Top row: shows an original image with the colour encoded scene flow. The bottom row shows different virtual viewing points on the same 3D scene. Left to right: panning to left with a tilt to the right, panning down to be in line with the road surface, panning up and titling down to be about 45° off horizontal, looking straight down from above the vehicle (birds-eye view).

eas become visible (e.g., the vehicle disappears off the image when it is very close to the camera). This issue is one that is difficult to avoid without flow occlusion detection, which is difficult to estimate by itself. The second problem that is common is seen at frame 210; the car turns left and has a lot of reflections on the windscreen. The vision algorithm detects the flow to the left (as the reflection goes this direction) but the true motion is to the right. Again, this is a common issue with optical flow and difficult to avoid.

The errors and standard deviation for the flow components u, v, p (Figure 16) are of similar shape, yet different magnitude. This is as expected, since they are solved using the same variational energy minimisation framework. The sequences are provided for public use and will allow comparisons of scene flow algorithms in future works.

5.4 Real-World Scenes

Real-world scenes are generally more complex to handle than synthetically generated images due to noise and artifacts that are not included in the model. We demonstrate the ability of the approach to deal with real-world scenes on images from a driver assistance system.

Figure 17 shows an example of the simplest case, where the camera is stationary. The scene flow reconstruction is very good with almost no outliers.

Figures 1, 18, 19, and 20 show scene flow results with a moving camera platform. Ego-motion of the camera is known from ego-motion estimation [2], using inertial sensors for the initial guess, and compensated in the depicted results.

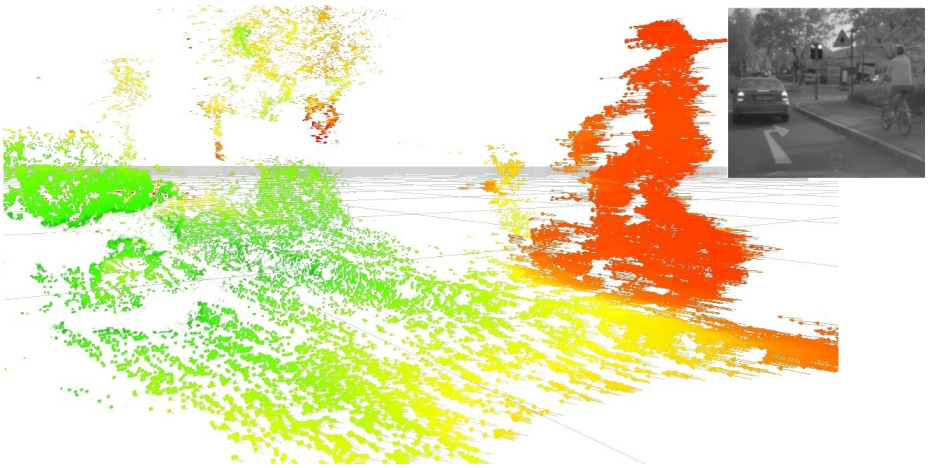


Fig. 19 Dense scene flow in a traffic scene.The colour in the scene flow image shows vector lengths after ego-motion compensation (green to red = 0 to $0.4m/s$). Only the cyclist is moving. The original image is in the upper right corner.

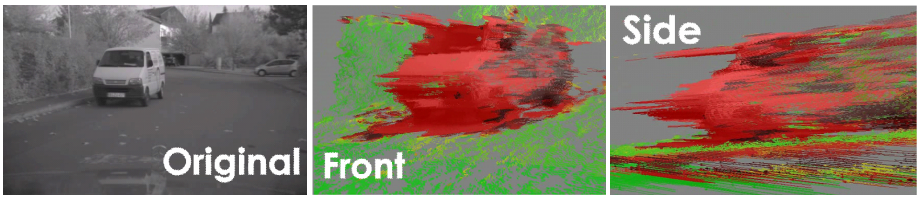


Fig. 20 A real-world example of our scene flow. The left image shows the original, the two other images show the scene flow reconstruction when viewed from the front and side. Colour encoding is green ↔ red is stationary ↔ moving.

Figure 1 shows results from a scene where a person runs from behind a parked vehicle. The ego-vehicle is driving forward at 30 km/h and turning to the left. The measurements on the ground plane and in the background are not shown to focus visual attention on the person. The results show that points on the parked vehicle are estimated as stationary, whereas points on the person are registered as moving. The accurate motion results can be well observed for the person's legs, where the different velocities of each leg are well estimated.

Figure 18 shows multiple virtual views of a vehicle that is followed by the ego-vehicle. This is to highlight that the vectors are clustered together and that the scene flow vectors are consistent.

Figure 19 shows an image from a sequence where the ego-vehicle is driving past a bicyclist. The depicted scene flow shows that most parts of the scene, including the vehicle stopping at the traffic lights, are correctly estimated as stationary. Only the bicyclist is moving and its motion is accurately estimated. Compared to Figure 17 there are more outliers in these results. This highlights that the ego-motion accuracy is vital when dealing with a moving platform, and slight errors are very noticeable in the results.

Figure 20 shows a van driving past the car. This figure demonstrates that the scene flow generates clustered vectors, all pointing to the same right direction with similar magnitude even when viewing from different angles.

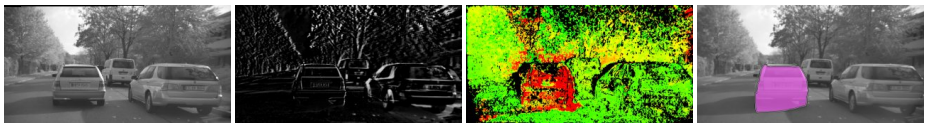


Fig. 21 From left to right: input image, difference image between two consecutive frames, residual motion likelihood, and segmentation result. With the residual motion likelihood derived from the scene flow, the segmentation of the moving object becomes possible although the camera itself is moving.

5.5 Segmentation using Uncertainties

Figure 21 shows an example of how the motion likelihood from Section 4.1 can be used as input to separate stationary from moving points in a segmentation task.

The segmentation is performed employing the graph cut segmentation algorithm from [4]. We consider individual uncertainty measures for the flow vectors and the disparities at each image pixel.

The segmentation algorithm requires to set up the graph $\mathscr{G}(n, n_s, n_t, e)$, consisting of nodes $n(x,y)$ for every pixel (x,y) in the reference image and two distinct nodes: the source node n_s and the target (sink) node n_t [21]. The edges e in this graph connect each node with the source, target, and its $\mathscr{N}_4$ neighbours (upper, lower, left, right). The individual edge costs are defined as follows:

edge e	edge cost
source link: $n_s \rightarrow n(x,y)$	$-\xi_{\text{motion}}(x,y)$
target link: $n(x,y) \rightarrow n_t$	$-\xi_{\text{static}}(x,y)$
neighbourhood: $n(\hat{x},\hat{y}) \leftrightarrow n(x,y)\ n(\hat{x},\hat{y}) \in \mathscr{N}_4$	$\beta \frac{1}{\lvert L(x,y) - L(\hat{x},\hat{y})\rvert + \alpha}$

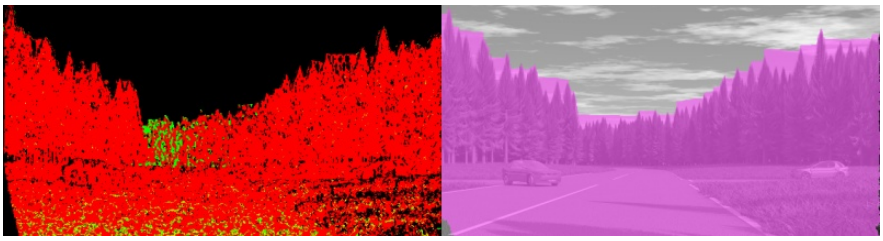
where ξ_{static} is a fixed value (globally equal likelihood of a point to be static), α is a small value to prevent numerical instability, β is a neighbourhood weighting function, and ξ_{motion} is a motion likelihood. For these results, $\xi_{\text{motion}} = \xi_M$ from Equation 23. For a more detailed explanation of the algorithm we refer to [40].

When using ξ_M for motion likelihood there are three possible ways that we can estimate the uncertainty. Using Equation 20:

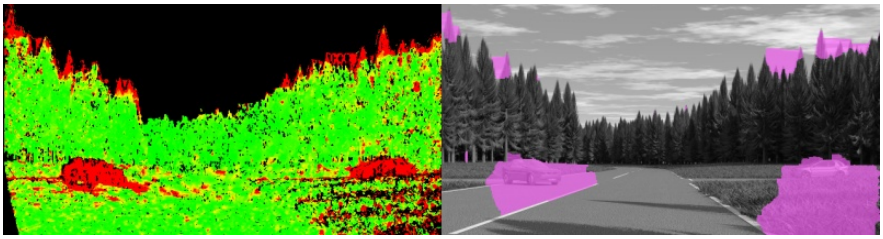
1. Assume that there is no error, i.e., $\Sigma_{SF} = \mathbf{I}$ (identity matrix).
2. Assume a spatially equal variance, i.e., $\sigma_\alpha = a$, where a is a constant.
3. Assume a pixel-wise local variance using uncertainty measures, i.e.,
   $\sigma_d = U_D$ (from Equation 15)
   $\sigma_u = \sigma_v = \sigma_p = U_{SF}$ (from Equation 16)

Figure 22 shows the difference in energy and segmentation when using the formulations above. The top image clearly shows that the approach using no error propagation fails. Using spatially fixed variances gives slightly better results (middle image). The best results are obtained with pixel-wise local variances estimated using a reasonable uncertainty model. This clearly shows that pixel-wise uncertainty estimation is very important. The outcome can probably be improved further by elaborating on the way to estimate the uncertainty. The rest of the results in this section are using the uncertainty measures in Equations 15 and 16 (assumption 3 above).

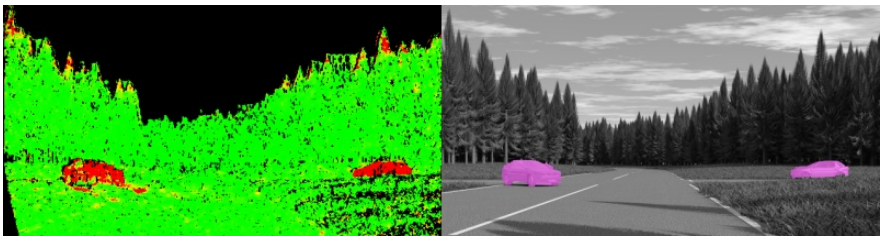
In a monocular setting, motion which is aligned with the epipolar lines cannot be detected without prior knowledge about the scene. Amongst other motion patterns, this includes objects moving parallel to the camera motion. For a camera moving in depth this



Assumption: no error



Assumption: spatially equal variances



Assumption: pixel-wise local variances from uncertainty measures

Fig. 22 Results for different error propagation methods. The *left* images show the residual motion likelihoods and the *right* images the segmentation results.

***PreceedingCar* scene**

***HillSide* scene**

Fig. 23 The figure shows segmentation of independently moving objects moving parallel to the camera movement. This movement cannot be detected monocularly without additional constraints, such as a planar ground assumption. Using our motion likelihood generates good segmentation results. Left to right: original image with segmentation results (pink), the optical flow image, and the energy image (green = low, red = high).

Fig. 24 The figure shows the energy images and the segmentation results for objects that do not move parallel to the camera motion. Note, that the also non-rigid independently moving objects are segmented. Colour coding as in Figure 23.

includes all (directly) preceding objects and (directly) approaching objects. Our scene flow motion likelihood provides good segmentation results, as seen in the *PreceedingCar* and *HillSide* sequences in Figure 23.

An example of non-rigid motion segmentation is shown in the *Running* sequence in Figure 24). This is to highlight the versatility of the scene flow motion likelihood.

6 Conclusions and Outlook

We have presented an efficient methodology to exploit information from both motion and stereo. We showed that decoupling the disparity estimation from the remainder of the estimation process is advantageous as it allows for selecting the most suitable methods for both tasks. This way, we were able to achieve higher accuracies at a lower computational cost. Furthermore, we presented a process for removing illumination differences between images, thus holding the intensity consistency assumption true. Finally, we proposed some uncertainty measures that worked well with movement segmentation.

Motion and disparity information are currently only rarely exploited for solving computer vision tasks. Research usually focuses on either motion or stereo, and in most cases neither motion or stereo is used. We believe that motion and stereo provide vital low-level information that must be taken into account to build reliable vision systems. In particular in the context of unsupervised techniques, depth and motion boundaries are key to separate and learn the appearance of independent objects. In this paper we showed some promising results on the segmentation of independent objects directly from scene flow. There is still much potential to exploit this information further.

Resolution	Pixels	Factor (px)	Time [ms]	Factor (Time)
320×240	$76,800$	1.0	48.2428	1.0
640×480	$307,200$	4.0	150.373	3.12
1280×960	$1,228,800$	16.0	572.911	11.88
1920×1440	$2,764,800$	36.0	1304.69	27.04

Fig. 25 The table indicades the real-time applicability of our algorithm if implemented on a modern GPU. The input images used (on different resolution scales) are shown below the table.

7 Appendix

7.1 Detailed Euler-Lagrange equations

The detailed Euler-Lagrange euqations derived in Section 2.4 result in the following linear equations:

$$\begin{aligned} &\Psi_{LF}^{'k,l} \cdot \left(E_{LF}^k + L_x^k\,\delta u^{k,l+1} + L_y^k\,\delta v^{k,l+1}\right) L_x^k \\ &+ c\,\Psi_{RF}^{'k,l} \cdot \left(E_{RF}^k + R_x^k\left(\delta u^{k,l+1} + \delta p^{k,l+1}\right) + R_y^k\,\delta v^{k,l+1}\right) R_x^k \\ &\qquad - \lambda\ \mathrm{div}\left(\Psi_S^{'k,l} \cdot \nabla\left(u^k + \delta u^{k,l+1}\right)\right) = 0 \end{aligned} \tag{31}$$

$$\begin{aligned} &\Psi_{LF}^{'k,l} \cdot \left(E_{LF}^k + L_x^k\,\delta u^{k,l+1} + L_y^k\,\delta v^{k,l+1}\right) L_y^k \\ &+ c\,\Psi_{RF}^{'k,l} \cdot \left(E_{RF}^k + R_x^k\left(\delta u^{k,l+1} + \delta p^{k,l+1}\right) + R_y^k\,\delta v^{k,l+1}\right) R_y^k \\ &\qquad - \lambda\ \mathrm{div}\left(\Psi_S^{'k,l} \cdot \nabla\left(v^k + \delta v^{k,l+1}\right)\right) = 0 \end{aligned} \tag{32}$$

$$\begin{aligned} &c\,\Psi_{RF}^{'k,l} \cdot \left(E_{RF}^k + R_x^k\left(\delta u^{k,l+1} + \delta p^{k,l+1}\right) + R_y^k\,\delta v^{k,l+1}\right) R_x^k \\ &\qquad + c\,\Psi_{DF}^{'k,l} \cdot \left(E_{DF}^k + R_x^k\,\delta p^{k,l+1}\right) R_x^k \\ &\qquad - \gamma\,\mathrm{div}\left(\Psi_S^{'k,l} \cdot \nabla\left(p^k + \delta p^{k,l+1}\right)\right) = 0 \end{aligned} \tag{33}$$

with

$$\Psi_*^{'k,l} := \Psi'\left(E_*\left(u^k + \delta u^{k,l}, v^k + \delta v^{k,l}, p^k + \delta p^{k,l}\right)\right) \tag{34}$$

Some terms from the original Euler-Lagrange equations have vanished due to the use of $R(x+d,y,t) = L(x,y,t)$ from the linearised disparity flow constraint (Eq. 4). After discretisation, the corresponding linear system is solved via successive over-relaxation (SOR) [45].

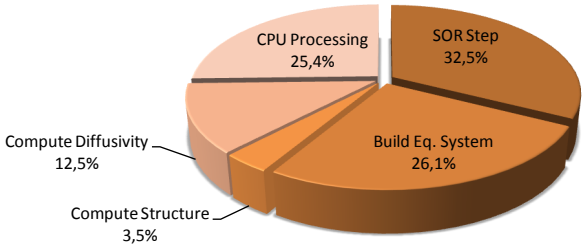


Fig. 26 Break down of computational time for our algorithm (3.0GHz Intel®Core™2 and NVidia®GeForce GTX 480) on 640×480 px images.

7.2 Implementation of Scene Flow

The scene flow algorithm was implemented in C++, obtaining a speed of 5 Hz on a 3.0GHz Intel®Core™2 CPU for QVGA images of 320×240 pixels. The implementation in CUDA for the GPU (NVidia®GeForce GTX 480) allows for a frame rate of 20 Hz as indicated in Figure 25. The settings for the computational times were 2 outer iterations (warps), 15 inner iterations, and 3 SOR iterations at each pyramid level. The parameters used are $\lambda = 0.06$, $\gamma = 0.6$, and $\omega = 1.99$ for the over-relaxation. Since we are interested in real-time estimates, we use only 4 levels in the pyramid, with a down-sampling rate of 0.5, i.e., each image is half the dimensions in both the x and y directions so $|\Omega|$ is cut down by 75% at each level. Although the energy on a smaller pyramid level is not exactly the same, it is a close approximation of the energy on the higher resolved images.

Figure 26 shows the break down of the computational time for our scene flow algorithm. Note, that the CPU processing includes memory management and the computation of the image pyramids, which offers some potential for optimization. The overview in pseudo-code for implementing the scene flow algorithm is shown in Algorithm 1. It calls subroutines described in Algorithms 2 to 5.

```
1:  for all levels do
2:      for all outer iterations do
3:          Compute Structure (Algorithm 2)
4:          Compute Diffusivity (Algorithm 3)
5:          u_tmp = u
6:          v_tmp = v
7:          p_tmp = p
8:          for all inner iterations do
9:              Build Equation System (Algorithm 4)
10:             for all SOR iterations do
11:                 SOR Step (Algorithm 5)
12:             end for
13:         end for
14:         Warp L(x,y,t) and R(x+d,y,t) using u,v and p.
15:     end for
16:     Warp u,v and p to upper level (double size and interpolate).
17: end for
```

Algorithm 1: Scene Flow Pseudo-Code

Algorithm 2 computes the spatial and temporal derivatives of the warped input images for the energy equations. The spatial derivatives are the average of the two contributing images. ∇ is computed using central differences and any reference outside Ω is clamped to the boundary (reflecting boundary conditions for the central difference operator).

The second algorithm within the outer loops, Algorithm 3, computes the diffusivities of the three-dimensional scene flow field. It consists of a combination of forward differences and central differences as proposed by Brox in [5]. All values outside Ω are clamped to the boundary (reflecting boundary conditions for the central difference operator).

Note, that this is computed once per warp before the current scene flow variables (u, v, p) are copied into temporal variables $(u_{\text{tmp}}, v_{\text{tmp}}, p_{\text{tmp}})$ to solve for the updates $(\delta u, \delta v, \delta p)$.

for all pixels **do**
  $\begin{bmatrix} L_x & L_y \end{bmatrix}^\top = \frac{1}{2}\left(\nabla L(x+u, y+v, t+1) + \nabla L(x, y, t)\right)$
  $L_t = L(x+u, y+v, t+1) - L(x, y, t)$
  if Disparity d known **then**
    $\begin{bmatrix} R_x & R_y \end{bmatrix}^\top = \frac{1}{2}\left(\nabla R(x+u+d+p, y+v, t+1) + \nabla R(x+d, y, t)\right)$
    $R_t = R(x+u+d+p, y+v, t+1) - R(x+d, y, t)$
    $D_x = \frac{1}{2}\left(\frac{\partial}{\partial x} R(x+u+d+p, y+v, t+1) + \frac{\partial}{\partial x} L(x+u, y+v, t+1)\right)$
    $D_t = R(x+u+d+p, y+v, t+1) - L(x+u, y+v, t+1)$
  else
    $\begin{bmatrix} R_x & R_y \end{bmatrix}^\top = 0$
    $D_x = 0$
  end if
end for

Algorithm 2: **Compute Structure**

for all pixels **do**
  for all $\alpha \in \{u, v, p\}$ **do**
    $R_{\alpha,\text{north}} = \left(\alpha(x,y) - \alpha(x,y\text{-}1)\right)^2 + \frac{1}{16}\left(\alpha(x\text{+}1,y) - \alpha(x\text{-}1,y) + \alpha(x\text{+}1,y\text{-}1) - \alpha(x\text{-}1,y\text{-}1)\right)^2$
    $R_{\alpha,\text{east}} = \left(\alpha(x,y) - \alpha(x\text{-}1,y)\right)^2 + \frac{1}{16}\left(\alpha(x,y\text{+}1) - \alpha(x,y\text{-}1) + \alpha(x\text{-}1,y\text{+}1) - \alpha(x\text{-}1,y\text{-}1)\right)^2$
    $R_{\alpha,\text{south}} = \left(\alpha(x,y\text{+}1) - \alpha(x,y)\right)^2 + \frac{1}{16}\left(\alpha(x\text{+}1,y\text{+}1) - \alpha(x\text{-}1,y\text{+}1) + \alpha(x\text{+}1,y) - \alpha(x\text{-}1,y)\right)^2$
    $R_{\alpha,\text{west}} = \left(\alpha(x\text{+}1,y) - \alpha(x,y)\right)^2 + \frac{1}{16}\left(\alpha(x\text{+}1,y\text{+}1) - \alpha(x\text{+}1,y\text{-}1) + \alpha(x,y\text{+}1) - \alpha(x,y\text{-}1)\right)^2$
  end for
  for all dir $\in \{\text{north}, \text{east}, \text{south}, \text{west}\}$ **do**
    $R_{\text{dir}} = \frac{\lambda}{\sqrt{R_{u,\text{dir}} + R_{v,\text{dir}} + \frac{\lambda^2}{\gamma^2} R_{p,\text{dir}}}}$
  end for
end for

Algorithm 3: **Compute Diffusivity**

However, instead of solving for the update and updating the original flow variables, the temporary variables are used inside Algorithm 4 to directly solve for the resulting flow field. Note instances of $u_{\text{tmp}} - u$, where the delta-updates are actually needed. We found that this *trick* speeds up the implementation of the flow field considerably.

Last, Algorithm 5 executes the inner iterations of the successive over-relaxation. On the GPU, this is implemented using the over-relaxed red-black Gauss-Seidel approach; see [33].

for all pixels **do**
  $E_{LF} = L_t + (u_{\text{tmp}} - u)L_x + (v_{\text{tmp}} - v)L_y$
  $\Psi'_{LF} = \frac{1}{\sqrt{E_{LF}^2 + \varepsilon^2}}$
  $E_{RF} = R_t + (u_{\text{tmp}} + p_{\text{tmp}} - u - p)R_x + (v_{\text{tmp}} - v)R_y$
  $\Psi'_{RF} = \frac{1}{\sqrt{E_{RF}^2 + \varepsilon^2}}$
  $E_{DF} = D_t + (p_{\text{tmp}} - p)D_x$
  $\Psi'_{DF} = \frac{1}{\sqrt{E_{DF}^2 + \varepsilon^2}}$
  $A_{uu} = \Psi'_{LF} L_x L_x + \Psi'_{RF} R_x R_x$
  $A_{uv} = \Psi'_{LF} L_x L_y + \Psi'_{RF} R_x R_y$
  $A_{vv} = \Psi'_{LF} L_y L_y + \Psi'_{RF} R_y R_y$
  $A_{up} = \Psi'_{RF} R_x R_x$
  $A_{vp} = \Psi'_{RF} R_x R_y$
  $A_{pp} = \Psi'_{RF} R_x R_x + \Psi'_{DF} D_x D_x$
  $b_u = \Psi'_{LF} L_x \left(L_t + u_{\text{tmp}} L_x + v_{\text{tmp}} L_y\right) + \Psi'_{RF} R_x \left(R_t + (u_{\text{tmp}} + p_{\text{tmp}})R_x + v_{\text{tmp}} R_y\right)$
  $b_v = \Psi'_{LF} L_y \left(L_t + u_{\text{tmp}} L_x + v_{\text{tmp}} L_y\right) + \Psi'_{RF} R_y \left(R_t + (u_{\text{tmp}} + p_{\text{tmp}})R_x + v_{\text{tmp}} R_y\right)$
  $b_p = \Psi'_{RF} R_x \left(R_t + (u_{\text{tmp}} + p_{\text{tmp}})R_x + v_{\text{tmp}} R_y\right) + \Psi'_{DF} D_x \left(D_t + p_{\text{tmp}} D_x\right)$
end for

Algorithm 4: **Build Equation System**

for all pixels **do**
  $R_{\text{sum}} = R_{\text{north}} + R_{\text{south}} + R_{\text{west}} + R_{\text{east}} + \varepsilon^2$
  $u_R = R_{\text{north}} u(x, y\text{-}1) + R_{\text{south}} u(x, y\text{+}1) + R_{\text{west}} u(x\text{-}1, y) + R_{\text{east}} u(x\text{+}1, y)$
  $u(x,y) = (1-\omega)u(x,y) + \frac{\omega}{A_{uu}+R_{\text{sum}}} \left(u_R - b_u - A_{uv} v(x,y) - A_{up} p(x,y)\right)$
  $v_R = R_{\text{north}} v(x, y\text{-}1) + R_{\text{south}} v(x, y\text{+}1) + R_{\text{west}} v(x\text{-}1, y) + R_{\text{east}} v(x\text{+}1, y)$
  $v(x,y) = (1-\omega)v(x,y) + \frac{\omega}{A_{vv}+R_{\text{sum}}} \left(u_R - b_u - A_{uv} u(x,y) - A_{vp} p(x,y)\right)$
  $p_R = R_{\text{north}} p(x, y\text{-}1) + R_{\text{south}} p(x, y\text{+}1) + R_{\text{west}} p(x\text{-}1, y) + R_{\text{east}} p(x\text{+}1, y)$
  $p(x,y) = (1-\omega)p(x,y) + \frac{\omega}{A_{pp}+R_{\text{sum}}} \left(u_R - b_u - A_{up} u(x,y) - A_{vp} v(x,y)\right)$
end for

Algorithm 5: SOR Step

References

1. Aujol, J.F., Gilboa, G., Chan, T., Osher, S.: structure-texture image decomposision – modeling, algorithms, and parameter selection. Int. J. of Computer Vision **67**(1), 111–136 (2006)
2. Badino, H.: A robust approach for ego-motion estimation using a mobile stereo platform. In: Proc. Int. Workshop on Complex Motion (IWCM04), pp. 198–208. Springer (2004)
3. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise smooth flow fields. Computer Vision and Image Understanding **63**(1), 75–104 (1996)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(9), 1124–1137 (2004)
5. Brox, T.: From pixels to regions: Partial differential equations in image analysis. Ph.D. thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany (2005). URL `www-cvpr.iai.uni-bonn.de/brox/pub/brox_PhDThesis.pdf`
6. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Proc. European Conf. on Computer Vision (ECCV), pp. 25–36. Springer (2004)
7. Bruhn, A., Weickert, J.: A confidence measure for variational optic flow methods. Geometric Properties for Incomplete Data **283-298**, Springer (2006)
8. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: Discontinuity preserving computation of variational optic flow in real-time. In: Proc. Int. Conf. on Scale-Space, pp. 279–290. Springer (2005)
9. Costeira, J., Kanande, T.: A multi-body factorization methd for motion analysis. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), pp. 1071–1076 (1995)
10. Franke, U., Joos, A.: Real-time stereo vision for urban traffic scene understanding. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 273–278. IEEE Computer Society, Dearborn (2000)

11. Gong, M.: Real-time joint disparity and disparity flow estimation on programmable graphics hardware. Computer Vision and Image Understanding **113**(1), 90–100 (2009)
12. Gong, M., Yang, Y.H.: Disparity flow estimation using orthogonal reliability-based dynamic programming. In: Proc. Int. Conf. on Pattern Recognition (ICPR), pp. 70–73. IEEE Computer Society (2006)
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521623049 (2000)
14. Hirschmüller, H.: Stereo vision in structured environments by consistent semi-globalmatching. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition(CVPR), pp. 2386–2393. IEEE Computer Society (2006)
15. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. on Pattern Analysis and Machine Intelligence **30**(2), 328–341 (2008)
16. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17**, 185–203 (1981)
17. Hu, X., Mordohai, P.: Evaluation of stereo con?dence indoors and outdoors. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition(CVPR), p. TBA. IEEE Computer Society (2010)
18. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), pp. 1–7. IEEE Computer Society (2007). URL `http://www-prima.imag.fr/prima/pub/Publications/2007/HD07/`
19. Isard, M., MacCormick, J.: Dense motion and disparity estimation via loopy belief propagation. In: Asian Conf. on Computer Vision (ACCV), *Lecture Notes in Computer Science*, vol. 3852, pp. 32–41. Springer (2006). URL `http://dblp.uni-trier.de/db/conf/accv/accv2006-2.html#IsardM06`
20. Kanatani, K., Sugaya, Y.: Multi-stage optimization for multi-body motion segmentation. IEICE Transactions on Information and Systems **E87-D**(7), 1935–1942 (2004)
21. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In: Proc. European Conf. on Computer Vision (ECCV), pp. 65–81. Springer (2002)
22. Mahalanobis, P.C.: On the generalised distance in statistics. In: Proc. of the National Institute of Science of India 12, pp. 49–55 (1936)
23. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Trans. on Image Processing **7**(5), 703–719 (1998)
24. Min, D., Sohn, K.: Edge-preserving simultaneous joint motion-disparity estimation. In: Proc. Int. Conf. on Pattern Recognition (ICPR), pp. 74–77. IEEE Computer Society, Washington, DC, USA (2006). DOI http://dx.doi.org/10.1109/ICPR.2006.470
25. Patras, I., Hendriks, E., Tziritas, G.: A joint motion/disparity estimation method for the construction of stereo interpolated images in stereoscopic image sequences. In: Proc. Int. Conf. on Pattern Recognition (ICPR), p. 359. IEEE Computer Society, Heijen, The Netherlands (1996)
26. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. of Computer Vision **72**(2), 179–193 (2007). DOI http://dx.doi.org/10.1007/s11263-006-8671-5
27. Rabe, C., Franke, U., Gehrig, S.: Fast detection of moving objects in complex scenarios. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 398–403. IEEE Computer Society (2007)
28. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition(CVPR) (2008)
29. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D. **60**, 259–268 (1992)
30. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondencealgorithms. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), pp. 7–42. IEEE Computer Society (2002)
31. Shimizu, M., Okutomi, M.: Precise sub-pixel estimation on area-based matching. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), pp. 90–97. IEEE Computer Society (2001)
32. Stein, F.: Efficient computation of optical flow using the census transform. In: Proc. DAGM (Pattern Recognition), pp. 79–86. Springer (2004)
33. Stüben, K., Trottenberg, U.: Multigrid methods: Fundamental algorithms, model problem analysis and applications, *Lecture Notes in Mathematics*, vol. 960. Springer (1982)
34. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University (1991). URL `citeseer.ist.psu.edu/tomasi91detection.html`
35. University of Auckland: *.enpeda..* Image Sequence Analysis Test Site (EISATS) (2008). `http://www.mi.auckland.ac.nz/EISATS/`
36. Vaudrey, T., Morales, S., Wedel, A., Klette, R.: Generalised residual images effect on illumination artifact removal for correspondence algorithms. Pattern Recognition **TBA**, TBA (2010). URL `http://dx.doi.org/10.1016/j.patcog.2010.05.036`

37. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: Proc. Int. Conf. on Image and Vision Computing New Zealand (IVCNZ). IEEE Computer Society (2008). IEEE Xplore:10.1109/IVCNZ.2008.4762133
38. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. IEEE Trans. on Pattern Analysis and Machine Intelligence **27**(3), 475 – 480 (2005)
39. Wedel, A., Pock, T., Zach, C., Cremers, D., Bischof, H.: An improved algorithm for TV-L1 optical flow. In: Revised Papers Int. Dagstuhl Seminar on Statistical and Geometrical Approaches to Visual Motion Analysis, pp. 23–45. Springer (2008)
40. Wedel, A., Rabe, C., Meissner, A., Franke, U., Cremers, D.: Detection and segmentation of independently moving objects from dense scene flow. In: D. Cremers, Y. Boykov, A. Blake, F.R. Schmidt (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition(EMMCVPR), vol. 5681, pp. 14–27. Springer (2009)
41. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Proc. European Conf. on Computer Vision (ECCV), pp. 739–751. Springer (2008)
42. Wedel, A., Vaudrey, T., Meissner, A., Rabe, C., Brox, T., Franke, U., Cremers20, D.: An evaluation approach for scene flow with decoupled motion and position. In: Revised Papers Int. Dagstuhl Seminar on Statistical and Geometrical Approaches to Visual Motion Analysis, pp. 46–69. Springer (2008)
43. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition(CVPR), p. TBA. IEEE Computer Society (2010)
44. Yan, J., Pollefeys, M.: A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Proc. European Conf. on Computer Vision (ECCV), *LNCS*, vol. 3954, pp. 94–106. Springer (2006)
45. Young, D.M.: Iterative Solution of Large Linear Systems. Academic Press, New York (1971)
46. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-L^1 optical flow. In: Proc. DAGM (Pattern Recognition), pp. 214–223. Springer (2007)
47. Zhang, Y., Kambhamettu, C.: On 3d scene flow and structure estimation. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition(CVPR), vol. 2, p. 778. IEEE Computer Society, Los Alamitos, CA, USA (2001). DOI http://doi.ieeecomputersociety.org/10.1109/CVPR.2001.991044
48. Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.P.: Complementary optic flow. In: Energy Minimization Methods in Computer Vision and Pattern Recognition(EMMCVPR), pp. 207–220. Springer (2009)