2014 Second International Conference on 3D Vision

aTGV-SF: Dense Variational Scene Flow through Projective Warping and Higher Order Regularization

David Ferstl, Christian Reinbacher, Gernot Riegler, Matthias Rüther and Horst Bischof
Graz University of Technology
Institute for Computer Graphics and Vision
Inffeldgasse 16, 8010 Graz, AUSTRIA
{ferstl,reinbacher,riegler,ruether,bischof}@icg.tugraz.at

Abstract

In this paper we present a novel method to accurately estimate the dense 3D motion field, known as scene flow, from depth and intensity acquisitions. The method is formulated as a convex energy optimization, where the motion warping of each scene point is estimated through a projection and back-projection directly in 3D space. We utilize higher order regularization which is weighted and directed according to the input data by an anisotropic diffusion tensor. Our formulation enables the calculation of a dense flow field which does not penalize smooth and non-rigid movements while aligning motion boundaries with strong depth boundaries. An efficient parallelization of the numerical algorithm leads to runtimes in the order of 1s *and therefore enables the method to be used in a variety of applications. We show that this novel scene flow calculation outperforms existing approaches in terms of speed and accuracy. Furthermore, we demonstrate applications such as camera pose estimation and depth image superresolution, which are enabled by the high accuracy of the proposed method. We show these applications using modern depth sensors such as Microsoft Kinect or the* PMD *Nano Time-of-Flight sensor.*

1. Introduction

Dynamic scene understanding is a relatively new research topic which tries to combine information from tracking, 3D reconstruction, segmentation and motion estimation to infer information about an ever changing 3D environment. In this paper we concentrate on the accurate measurement of motion in space, known as Scene Flow (*SF*), which is a key ingredient for dynamic scene understanding. Traditionally, motion estimation in computer vision has been addressed under the name of Optical Flow (*OF*), starting with the seminal work of Horn and Schunk [11]:

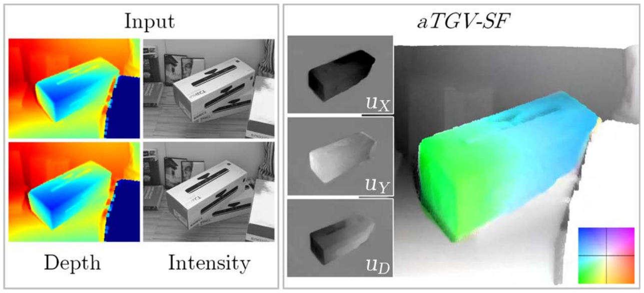


Figure 1. *SF* estimation. Two consecutive depth and intensity acquisitions are used to estimate *SF* in a variational energy minimization framework using anisotropic Total Generalized Variation (*aTGV-SF*). The X, Y and Z components of the *SF* are visualized as gray code. For better visualization also the color coded X and Y components are overlaid on the corresponding depth map.

The goal of *OF* estimation is to recover an accurate motion in 2D image space. Since *OF* only estimates a 2D projection of the 3D motion, it has problems to recover the motion in space. Relatively ferdrple 3D movements such as a translation towards the camera produce complex 2D motions which are hard to be estimated by current *OF* approaches. *SF* can be estimated by combining traditional *OF* with depth reconstruction in a calibrated and synchronized multi-view setup, as shown in [1, 26, 28].

Recently, once prohibitively expensive range sensors became available to the mass market with the introduction of Microsoft Kinect, Intel Gesture Camera or a variety of other Time of Flight (*ToF*) cameras. The usage of these sensors allows to dispense with multi-view depth reconstruction, instead directly accessing *dense* depth data.

The proposed approach combines depth and intensity data to calculate a dense *SF* field. In contrast to many approaches we define the warping functions directly by projection and back-projection in 3D space using the standard projective camera model together with the known depth. A higher order regularization term, namely Total Generalized Variation (*TGV*), is able to model smooth and non-rigid mo-

978-1-4799-7000-1/14 $31.00 © 2014 IEEE
DOI 10.1109/3DV.2014.19

tion alike. We further build on the observation that motion boundaries are more likely to appear at depth discontinuities, while regions with less variations in the structure are more likely to have similar motion vectors. Hence, the input depth information is used to weight and direct the regularization by an anisotropic diffusion tensor. Fusing the information of both depth and intensity data, an accurate and dense 3D motion is calculated. Furthermore, our model captures smooth flow transitions which occur at rotating objects and non-rigid movements. Nevertheless, sharp boundaries of the flow field between objects can still be preserved.

The main contributions of this work are threefold: 1) We build a *SF* model with a depth and an intensity image constraint where the temporal difference is calculated as a projection and back-projection in 3D space. 2) We use higher order regularization together with anisotropic diffusion based on the input data to better handle rotations and non-rigid movements. 3) We formulate the proposed model as a convex energy minimization problem which can be parallelized efficiently, enabling the application of our method to a variety of problems.

In numerical and visual comparisons to state of the art (*SOTA*) approaches we show that our method is superior in terms of speed and accuracy. We further demonstrate applications enabled by the high accuracy of our method, such as camera localization and depth image superresolution.

2. Related Work

Originating from the seminal work of Horn and Schunk [11] a vast amount of work has been done on *OF* estimation. Recently, Sun *et al*. [25] surveyed the different *OF* approaches and their principles. Although these methods can be coupled with 3D images to calculate *SF* they do not utilize this information during optimization. In the emerging field of *SF* calculation, existing methods can be mainly divided into the estimation from a sequential acquisition in a fully calibrated multi-view system and the calculation of *SF* from the sequential acquisition of depth and intensity data. Since our method clearly belongs to the latter, we will mainly focus the literature review on this category.

The first definition of the terminology of *SF* was given by Vedula *et al*. [26]. They calculate *OF* in a Lukas Kanade (*LK*) [16] framework per camera and fit the *SF* to the *OF*. Following this multi-view approach a lot of follow up work has been done, such as [1, 3, 14, 22, 27, 28].

Enabled by recently affordable depth sensors *SF* methods based on the combination of depth and intensity data have been proposed. Hadfield and Bowden proposed a method where the *SF* problem is modeled using a particle filter to avoid over-smoothing in the flow field [9]. The approach ranks among local methods since the result is a sparse *SF* field. Similarly, Quiroga *et al*. [20] estimate *SF* locally in a *LK* framework, later they extend this method by embedding it in a dense optimization framework [21].

Letouzey *et al*. [15] cast the optimization as a linear problem which can be solved very efficiently. Visual features on intensity images like SIFT are combined with an extension of [11] on depth images. Gottfried [8] *et al*. propose a complete framework for the calibration, *OF* and range flow estimation specifically tailored towards the Microsoft Kinect sensor. Zhang *et al*. [30] combine a global energy minimization with a bilateral filter to detect occlusion caused by imaging hardware in a two-step framework. A generalization of variational *OF* algorithms for *SF* estimation has been shown by Herbst *et al*. [6]. They further showed how *SF* can help to better segment objects from motion. Hornáček *et al*. [12] recently showed the advantages of estimating 3D motion directly through a *RGB-D* patch matching in the point cloud.

Existing approaches on *SF* estimation from a sequence of corresponding intensity and depth images can be divided into local and global methods. Local methods like [9, 20] are only able to estimate a sparse *SF* field. Global methods on the other hand are able to deliver a dense flow field by interpolating the missing data using some regularization method. Our method builds on the success of global optimization methods like [15, 6, 8, 30]. While most of them separate *SF* into 2D *OF* with an additional depth factor, our method calculates dense *SF* directly in 3D space. By defining warping functions as projection and back-projection in 3D space, our method estimates all components of *SF* jointly. Our method inherently handles the effect of object magnification in image space which is caused by object movement towards or away from the camera.

Instead of relying on first order regularization with a *L2* or Charbonnier penalty, we use a higher order regularization with *L1* penalization. Motion boundaries between objects are preserved as well as smooth transitions caused by rotation or non-rigid movements. Compared to first order regularization, our method avoids stair-casing and flow-flattening. Based on the observation that motion and object boundaries often coincide, we use an anisotropic diffusion tensor on the depth image that weights and orients the flow gradient during the optimization process.

3. Method

SF estimation is concerned with the motion of 3D scene points observed at two different time instances. Consider the consecutive acquisition of a scene at time instances $t = \{1, 2\}$ resulting in two depth and intensity image pairs D_1, I_1 and $D_2, I_2 \colon (\Omega \subseteq \mathbb{R}^2) \mapsto \mathbb{R}$. The motion of each scene point $\mathbf{X} = [X, Y, D]^T$ is given by $\mathbf{u} = \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = [u_X, u_Y, u_D]^T$. With that we can define the movement of

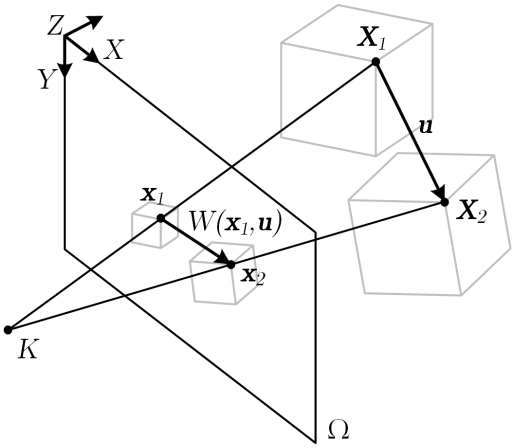


Figure 2. Flow Geometry. A scene point $\mathbf{X}_1$ acquired in the first frame moves to $\mathbf{X}_2$ in the second frame. This 3D movement between two acquisitions is defined as flow $\mathbf{u}$. The projection in the image space from point $\mathbf{x}_1$ to $\mathbf{x}_2$ is defined as the warping $W(\mathbf{x}_1, \mathbf{u})$.

$\mathbf{X}$ from time $t = 1$ to $t = 2$ as

$$\begin{aligned} \mathbf{X}_2 \quad &= \quad \mathbf{X}_1 \quad + \quad \mathbf{u}, \\ \begin{bmatrix} X \\ Y \\ D \end{bmatrix}_2 &= \begin{bmatrix} X \\ Y \\ D \end{bmatrix}_1 + \begin{bmatrix} u_X \\ u_Y \\ u_D \end{bmatrix}. \end{aligned} \tag{1}$$

The objective of *SF* is to find the dense flow field $\mathbf{u} : \Omega \mapsto \mathbb{R}^3$, which minimizes an image (4) and a depth error (6) criterion. To solve the ill-posed problem we add constraints on noise and constancy expressed as a regularization force. We first describe the general geometric flow model between two frames in 3.1. With this definition, the frame to frame energy minimization model is shown in 3.2. The implementation details to solve this minimization objective are given in 3.3.

3.1. Scene Flow Formulation

The scene points are observed through projections at image positions $\mathbf{x} = [x, y]^T \in \Omega$ with depth $D(\mathbf{x})$. Each scene point is therefore given by $\mathbf{X} = K^{-1}\mathbf{x}^h D(\mathbf{x})$, where K is the camera projection matrix and h denotes homogeneous image coordinates. Hence, the movement between two frames in image space is defined by

$$\mathbf{x}_2^h = W(\mathbf{x}_1, \mathbf{u})^h = \frac{K(K^{-1}\mathbf{x}_1^h D_1(\mathbf{x}_1) + \mathbf{u})}{D_1(\mathbf{x}_1) + u_D}. \tag{2}$$

This geometric relationship is depicted in Fig. 2.

The image warping in 3D space enables to define the brightness difference in the intensity image domain w.r.t. time, equal to the brightness constancy constraint in *OF* estimation,

$$I_{\Delta t}(\mathbf{x}_1, \mathbf{u}) = I_2(W(\mathbf{x}_1, \mathbf{u})) - I_1(\mathbf{x}_1), \tag{3}$$

which should be zero for noise-free input data and an optimal flow. We use a linear approximation of the brightness difference at a given initial flow field $\mathbf{u}_0$ applying a first order Taylor expansion which yields the intensity constraint

$$\begin{aligned} \rho_I(\mathbf{x}_1, \mathbf{u}, c) = I_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) + \nabla I_2(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0}(\mathbf{u} - \mathbf{u}_0) \\ + \delta c(\mathbf{x}_1), \end{aligned} \tag{4}$$

where W_0 is an abbreviation for $W(\mathbf{x}_1, \mathbf{u}_0)$. $I_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0)$ is the brightness difference (3) evaluated at $\mathbf{u}_0$. In most intensity acquisitions, shadows, specular highlights or slight illumination changes occur. To compensate the violation of the brightness constancy, we incorporate a compensation variable $c(\mathbf{x})\colon (\Omega \subseteq \mathbb{R}^2) \mapsto \mathbb{R}$ in our model, according to [5]. The parameter $\delta \in \mathbb{R}$ steers the influence of the compensation.

A similar constraint has to be fulfilled also by the depth data. According to the flow definition (1) the depth difference results in

$$D_{\Delta t}(\mathbf{x}_1, \mathbf{u}) = D_2(W(\mathbf{x}_1, \mathbf{u})) - D_1(\mathbf{x}_1) - u_D, \tag{5}$$

which is again zero for an optimal *SF* estimation as shown in Fig. 2. The depth constraint in our optimization model is given by a first order Taylor expansion of (5):

$$\rho_D(\mathbf{x}_1, \mathbf{u}) = D_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) + \nabla D_2(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0}(\mathbf{u} - \mathbf{u}_0). \tag{6}$$

These two *SF* constraints (4) and (6) are the data terms which lead to our convex optimization model.

3.2. Energy Optimization

We calculate a dense flow field through convex optimization. In a global model, both the intensity (4) and the depth constraint (6) provide just two constraints for three unknowns at each pixel (u_X, u_Y, u_Z). To solve such an ill-posed problem, a common way is to introduce a regularization term. Popular regularization terms in *OF* and *SF* methods are composed of first order regularizers with *L1* or *L2* norms or non-local variations thereof. *L2* is quite easy to minimize but often leads to over-smoothed results while *L1* enforces piecewise constant solutions. Utilizing a higher order model, namely the *TGV* introduced by Bredies *et al.* [2], we are able to obviate the problems of first order methods. The *TGV* allows for a reconstruction of piecewise polynomial functions.

The primal definition of the second order *TGV* is formulated as

$$TGV_{\alpha}^{2} = \min_{\mathbf{u},\mathbf{v}} \left\{ \alpha_1 \int_{\Omega} |\nabla \mathbf{u} - \mathbf{v}| \, \mathrm{d}x + \alpha_0 \int_{\Omega} |\nabla \mathbf{v}| \, \mathrm{d}x \right\}, \tag{7}$$

where additionally to the first order smoothness, the auxiliary variable $\mathbf{v}$ is introduced to enforce second order smoothness. α_0, $\alpha_1 \in \mathbb{R}$ are weighting parameters to balance the first and second order terms respectively. Because the *TGV* regularizer is convex it allows to compute a globally optimal solution.

TGV already shows edge preserving capabilities. However, to improve the resulting flow field around strong depth borders, we additionally weight the regularization term with gradient information from the input depth image D_1. We include an anisotropic diffusion tensor $T^{\frac{1}{2}}$, known as the Nagel-Enkelmann operator [17] which is inspired by the recent success in using anisotropically weighted *TGV* for 3D reconstruction [23] and depth map upsampling [7]. This tensor is calculated by

$$T^{\frac{1}{2}} = \exp\left(-\beta \, |\nabla D_1|^{\gamma}\right) \mathbf{n}\mathbf{n}^{\mathrm{T}} + \mathbf{n}^{\perp}\mathbf{n}^{\perp\mathrm{T}}, \tag{8}$$

where $\mathbf{n} = \frac{\nabla D_1}{|\nabla D_1|}$ is the normalized direction of the depth image gradient and $\mathbf{n}^{\perp}$ is the normal vector to the gradient. The gradients are calculated using the Sobel operator to reduce the influence of noise in the tensor. The scalars $\beta, \gamma \in \mathbb{R}$ adjust the magnitude and the sharpness of the tensor.

The final energy in our optimization model is composed of the *L1*-penalized intensity (4) and depth constraint (6) and the *TGV* regularization term (7) with anisotropic diffusion:

$$\min_{\mathbf{u},\mathbf{v},c} \left\{ \lambda_I \int_{\Omega} w|\rho_I| \, \mathrm{d}x + \lambda_D \int_{\Omega} w|\rho_D| \, \mathrm{d}x + \int_{\Omega} |\nabla c|_{\epsilon} \, \mathrm{d}x \right.$$
$$\left. + \, \alpha_1 \int_{\Omega} |T^{\frac{1}{2}}(\nabla \mathbf{u} - \mathbf{v})| \, \mathrm{d}x + \alpha_0 \int_{\Omega} |\nabla \mathbf{v}| \, \mathrm{d}x \right\}. \tag{9}$$

The pixelwise confidence score $w\colon \Omega \mapsto [0,1]$ can be derived from the depth sensor if applicable. This confidence is set to 0 where no depth measurements are available, *e.g.* for stereo sensors at occluded regions. The illumination model c is expected to be smooth, we therefore regularize it with the Huber norm [13] parameterized by ϵ.

3.3. Implementation

In this section we will detail the numerical implementation of the optimization method in (9). (9) is convex but clearly non-smooth. We solve this optimization problem by introducing Langrange multipliers for the constraints in (6) and (4) and applying a Legendre Fenchel transform (*LF*). With that, the problem can be reformulated as the convex-concave saddle point problem discretized on a Cartesian grid of size $M \times N$

$$\min_{\mathbf{u},\mathbf{v},c} \max_{\mathbf{p}_{\mathbf{u}},\mathbf{p}_{\mathbf{v}},p_c,q_D,q_I} \lambda_I \left\langle w\rho_I, q_I \right\rangle_{Q_I} + \lambda_D \left\langle w\rho_D, q_D \right\rangle_{Q_D}$$
$$+ \, \alpha_1 \left\langle T^{\frac{1}{2}}(\nabla \mathbf{u} - \mathbf{v}), \mathbf{p}_{\mathbf{u}} \right\rangle_{P_{\mathbf{u}}} + \alpha_0 \left\langle \nabla \mathbf{v}, \mathbf{p}_{\mathbf{v}} \right\rangle_{P_{\mathbf{v}}}$$
$$+ \left\langle \nabla c, \mathbf{p}_c \right\rangle_{P_c} + \frac{\varepsilon \|\mathbf{p}_c\|_2^2}{2}, \tag{10}$$

where the convex sets for the dual variables result in

$$\begin{aligned}
P_{\mathbf{u}} &= \left\{ \mathbf{p}_{\mathbf{u}}\colon \Omega \to \mathbb{R}^{6MN} \,\middle|\, \|\mathbf{p}_{\mathbf{u}}(i,j)\| \leq 1 \right\}, \\
P_{\mathbf{v}} &= \left\{ \mathbf{p}_{\mathbf{v}}\colon \Omega \to \mathbb{R}^{12MN} \,\middle|\, \|\mathbf{p}_{\mathbf{v}}(i,j)\| \leq 1 \right\}, \\
P_c &= \left\{ \mathbf{p}_c\colon \Omega \to \mathbb{R}^{2MN} \,\middle|\, \|\mathbf{p}_c(i,j)\| \leq 1 \right\}, \\
Q_D &= \left\{ q_D\colon \Omega \to \mathbb{R}^{MN} \,\middle|\, -1 \leq q_D(i,j) \leq 1 \right\}, \\
Q_I &= \left\{ q_I\colon \Omega \to \mathbb{R}^{MN} \,\middle|\, -1 \leq q_I(i,j) \leq 1 \right\}, \\
i &= 1, ..., M, j = 1, ..., N.
\end{aligned} \tag{11}$$

We can now apply the primal-dual minimization scheme proposed in [4] to solve (10). In this optimization scheme it is possible to solve for *L1* norms instead of using a *L2* norm or a differentiable Charbonnier *L1* approximation, as used in other methods. In addition, it can be efficiently parallelized which results in high frame rates. Due to lack of space, we will outline the complete numerical scheme in the supplementary material. The linearization of the *SF* constraints (4) and (6) is only valid for small displacements in the pixel level. Therefore the optimization has to be embedded into a coarse-to-fine framework. We employ image pyramids with a downsampling factor of $\nu = 0.8$ for this purpose. Due to warping in 3D space the camera projection matrix has to be adjusted accordingly. The weighting parameters for all terms in our energy are kept constant over all levels.

4. Evaluation

In this section we provide an extensive qualitative and quantitative evaluation of our method, which we address as *aTGV-SF*. For the real world evaluations we used a *PMD* Nano *ToF* camera [19] with a resolution of 120 × 160 and a *Microsoft Kinect for Windows v2 camera (K4Wv2)* with a depth image resolution of 512 × 424[a]. As error measurements we consistently use the Average Angular Error (*AAE*), Average End Point Error (*EPE*) and Root Mean Squared Error of flow in z (RMS_{V_z}). *AAE* and *EPE* are calculated in 2D (with subscript OF) and 3D (subscript SF).

[a] The K4Wv2 developer kit is preliminary software and/or hardware and APIs are preliminary and subject to change.

Since the choice of the parameters in our optimization model depends on the sensor and the application, they are manually set once and kept constant for each experiment. The parameters δ, ϵ for illumination compensation are set to 0.01, 0.01 and the tensor parameters β, γ are set to 10.0, 0.8 for all experiments. The reported execution time of our method is measured as mean over 100 runs computed on a recent PC with a Nvidia GTX680 GPU. Further visual evaluations are shown in the supplemental material.

4.1. Scene Flow Evaluation on Synthetic and Real Datasets

This experiment shows the different properties and contributions of the individual terms in our *SF* model. Besides that, we show a comparison to *SOTA OF* and *SF* methods. For a quantitative evaluation we create a synthetic dataset where a cube is translated and rotated in a defined way in front of a static background. This dataset includes the groundtruth *SF*, the input depth image pairs as well as the intensity image pairs where a marbled texture was applied to the 3D scene and rendered from the same view point as the depth images. We further added Gaussian noise on the input data to simulate acquisition noise. The errors are shown for a pure translation of 20% of the object size in X direction (T_X), a pure translation of 20% towards the camera (T_Z) and a rotation of 15 degrees about the Z axis (R_Z). An example is shown in Fig. 3. In Table 1, the results are shown compared to two *OF* methods, namely and the *NL-TV-NCC* methods of Werlberger *et al.* [29] and the *Classic-NL-Full* method of Sun *et al.* [25] using their publicly available code. We further compare it to one *SOTA SF* method, namely *SphereFlow* from Hornáček *et al.* [12]. To calculate the error measures in 2D, the result of our method is back-projected to image space, while for 3D errors the result of the *OF* methods is projected into 3D space with the noise free depth map.

A qualitative evaluation for different camera modalities for freely moving rigid and non-rigid objects is shown in Fig. 6. The average runtime of our method is 0.45s for the synthetic experiment, 0.84s for *PMD Nano* and 1.63s for *K4Wv2* per image.

This experiment points out the properties of the different terms in our model. While a first order model (*TV-reg*) works well for constant translational movements in any direction it is not suitable to model smooth flow transitions *e.g.* rotations or non-rigid movements since it forces piecewise constant solutions in the flow field. The anisotropic tensor has a big impact on the quality of the estimation since it directly bounds the *TGV* regularization along the object boundaries. The *RGB-D* patch-match based method from Hornáček *et al.* [12] (*SphereFlow*) delivers comparable results for the flow magnitude (*EPE*) but lacks in angular precision (*AAE*) in 3D. One problem could be observed at higher noise noise levels or illumination changes as shown in the *PMD* Nano sequences in Fig. 6.

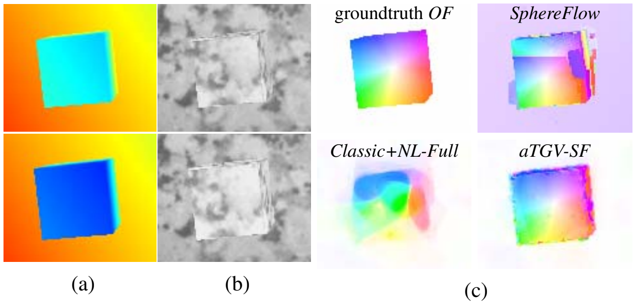


Figure 3. Flow estimation on a synthetic datasets. In (a) the color-coded input depth and in (b) the input intensity images are shown. The groundtruth *OF* and the results for this pure Z movement are shown in (c). While the pure *OF* methods estimate a flow field with increasing magnitude from the center to the border, our *SF* only estimates constant flow in Z which results in a better movement estimation.

Using our method for *OF* estimation shows comparable results to *SOTA OF* methods for object movements parallel to the image plane. The real advantage of our model emerge for object movements in depth, as shown in Fig. 3. While a constant movement in the Z direction is easy to estimate with our method in three dimensions, it is very hard to estimate in just two dimensions because this results in an *OF* with divergent motion in image space.

4.2. Middlebury Evaluation

The overall performance of our method compared to *SOTA SF* methods was evaluated using an existing *SF* benchmark dataset. We follow [1, 14, 9, 20, 21, 30], which use the rectified stereo intensity and disparity maps from the Middlebury *Cones*, *Teddy* and *Venus* datasets [24] to create *SF*. The depth maps are calculated using a defined baseline. The pure horizontal camera movement of the stereo intensity pair allows to calculate the ground truth scene motion at every point in the scene with a X movement given by the baseline while the movement in Y and Z direction is zero. The estimated *SF* is back-projected into the image space for a direct comparison with the ground truth disparity maps. In Table 2 the evaluation results compared to several *SOTA* methods for *SF* from stereo and *SF* from depth and intensity data are shown. Our method consistently delivers a *SF* quality which is superior to the compared *SOTA* methods, while being computationally efficient. It should be noted that the methods of [1, 14] both solve a harder problem since they do not utilize the noise-free disparity maps but compute stereo matches and *SF* jointly. Both methods are listed here for the sake of completeness. To show the general applicability we deliberately use the same parameters for all three datasets

	$T_X = 20\%$		$T_Z = -20\%$		$R_Z = 15°$	
	EPE$_{\text{OF}}$ / *AAE*$_{\text{OF}}$	*EPE*$_{\text{SF}}$ / *AAE*$_{\text{SF}}$	*EPE*$_{\text{OF}}$ / *AAE*$_{\text{OF}}$	*EPE*$_{\text{SF}}$ / *AAE*$_{\text{SF}}$	*EPE*$_{\text{OF}}$ / *AAE*$_{\text{OF}}$	*EPE*$_{\text{SF}}$ / *AAE*$_{\text{SF}}$
NL-TV-NCC [29]	0.421 / 6.31	0.282 / 5.61	0.130 / 5.49	0.191 / 3.07	0.467 / 5.75	0.291 / 5.06
Classic+NL-Full [25]	**0.252** / **3.08**	0.260 / 4.57	0.143 / 5.61	0.303 / 3.29	0.495 / 5.21	0.388 / 5.68
SphereFlow [12]	0.404 / 13.60	0.089 / 3.85	0.221 / 11.10	0.090 / 3.30	$\underline{0.224}$ / 8.00	$\underline{0.056}$ / 2.43
aTGV-SF TV-reg	$\underline{0.294}$ / $\underline{4.94}$	**0.064** / **2.40**	$\underline{0.094}$ / $\underline{3.95}$	$\underline{0.088}$ / 1.88	0.276 / $\underline{5.35}$	0.172 / $\underline{2.09}$
aTGV-SF w/o tensor	0.375 / 6.99	0.081 / 3.14	0.197 / 5.85	0.185 / $\underline{1.40}$	0.323 / 5.80	0.066 / 2.10
aTGV-SF	0.302 / 5.78	$\underline{0.066}$ / $\underline{2.54}$	**0.091** / **3.54**	**0.085** / **1.26**	**0.211** / **4.96**	**0.048** / **1.83**

Table 1. *SF* evaluation on a synthetic dataset. Comparison of our method with state of the art *OF* methods at different object movements in terms of *EPE* and *AAE* in 2D and 3D. Further, evaluation results of our methods are shown, where different terms are turned off. The best result for each movement is highlighted and the second best is underlined.

	Cones			*Teddy*			*Venus*			Avg.Time [s]
	EPE$_{\text{OF}}$	/ *RMS*$_{\text{Vz}}$	/ *AAE*$_{\text{OF}}$	*EPE*$_{\text{OF}}$	/ *RMS*$_{\text{Vz}}$	/ *AAE*$_{\text{OF}}$	*EPE*$_{\text{OF}}$	/ *RMS*$_{\text{Vz}}$	/ *AAE*$_{\text{OF}}$	
Basha et al. [1](2 views) (st)	0.58	N/A	0.39	0.57	N/A	1.01	0.16	N/A	1.58	-
Huguet and Devernay [14] (st)	1.10	N/A	0.69	1.25	N/A	0.51	0.31	N/A	0.98	5h
Hadfield and Bowden [10]	1.24	0.06	1.01	0.83	0.03	0.83	0.36	$\underline{0.02}$	1.03	10min
Quiroga et al. [21]	0.57	0.05	$\underline{0.52}$	0.69	0.04	0.71	0.31	**0.00**	1.26	$\underline{\text{10s}}$
Hornáček et al. [12]	$\underline{0.54}$	**0.02**	$\underline{0.52}$	$\underline{0.35}$	$\underline{0.01}$	$\underline{0.16}$	$\underline{0.26}$	$\underline{0.02}$	$\underline{0.64}$	-
aTGV-SF	**0.35**	$\underline{0.03}$	**0.04**	**0.09**	**0.00**	**0.01**	**0.06**	**0.00**	**0.27**	**2.4s**

Table 2. Quantitative comparison of *SF* methods on the Middlebury dataset. The error is measured by *EEP* and *AAE* in 2D. The best result for each dataset is highlighted and the second best is underlined. Methods that calculate SF from stereo are marked with (st).

of the Middlebury even though the Venus dataset has other lighting and surface conditions.

5. Applications

Fast and accurate *SF* estimation has many potential computer vision applications. In this section we will present two applications of our *SF* estimation method on real-world data. In 5.1 we show how to estimate the camera pose in a static scene without explicitly building a model of the scene. In 5.2 we use our model to increase the lateral resolution of a depth image by moving an object in front of an observing depth camera.

5.1. Camera Pose Estimation

An accurate *SF* enables the application of estimating the pose of a moving camera in space in a static scene. Since our method calculates metric *SF* we can directly estimate the movement of each scene point from one frame to the next. Given the estimated flow field $\mathbf{u}$ between two consecutive frames $t = \{1, 2\}$, we can establish corresponding point sets $\mathbf{X}_1$ and $\tilde{\mathbf{X}}_2 = \mathbf{X}_1 + \mathbf{u}$. As in traditional pose estimation, the general rotation $R_1 \in \mathrm{SO}(3)$ and translation $T_1 \in \mathbb{R}^3$ is calculated by Euclidean motion estimation as $\min_{R_1,T_1} \left(R_1\mathbf{X}_1 + T_1 - \tilde{\mathbf{X}}_2\right)^2$. The camera pose is updated by $P_2 = P_1[R_1|T_1]^{-1}$, where P_1 and P_2 are the camera poses. For multiple frames this pose estimation can be propagated by $P_{t+1} = P_{t_1} \bigcap_{i=t_1}^{t} [R_i|T_i]^{-1}$, where t_1 denotes the first frame.

For the numerical evaluation of the camera pose estimation, we use the PMD Nano *ToF* camera mounted on the head of an industrial robot. We use the real scene from Fig. 6 (2$^{\text{nd}}$ row) acquired by different, known camera poses instead of moving the objects. The robot moves 260mm in a linear movement in positive X, Y and Z direction with a distance of 1.73, 8.66 and 17.32mm between consecutive acquisitions. The sequence and intermediate results can be found in the supplementary video. The estimation accuracy is compared to standard *ICP* with 100 iterations and to a model based multi-scale *ICP* as proposed in the *KFusion* framework [18]. The camera position of the first frame P_{t_1} is defined as the world coordinate center. To quantify the error, the metric difference between the accumulated camera poses P_t and the known robot poses is calculated in terms of Root Mean Squared Error (*RMSE*) and shown in Fig. 4. The mean error per mm movement in $X/Y/Z$ direction is

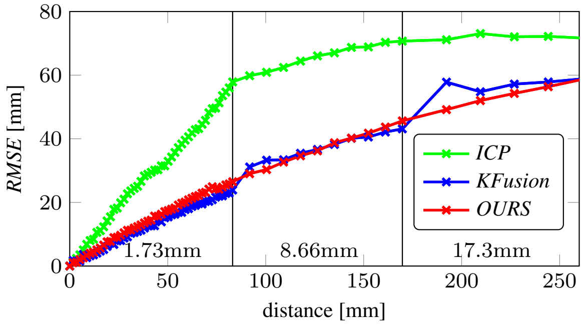


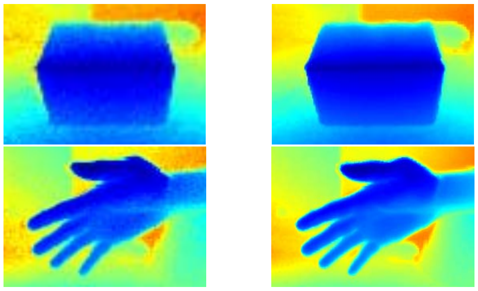
Figure 4. Evaluation of camera pose estimation using our method (*aTGV-SF*), standard *ICP* and the model based *ICP* (*KFusion*). The error is given by means of *RMSE* between real and estimated camera pose in [mm] for a relative distance between two consecutive frames of 1.73, 8.66 and 17.32mm.

$0.35/0.13/0.30$ for *ICP*, $0.09/0.3/0.23$ for *KFusion* compared to $0.07/0.06/0.36$ with our method. The relative rotation error in the pose estimation is below 0.8 degrees for all three methods. Because we assume a static scene, the *SF* computation is accelerated by reducing the number of levels and iterations per level resulting in an average frame rate of 15.3fps.

In the error statistics of mean and relative error it can be seen that *KFusion* and our method clearly outperform standard *ICP* used for camera pose estimation. Further, it can be seen that the relative error is not dependent on the movement magnitude. Compared to the model based multi-scale *KFusion* we achieve comparable results without the need of keeping an explicit model of the scene.

5.2. Superresolution

Similar to the camera pose estimation, *SF* can also be used for depth superresolution of a scene. In this experiment we show how our *SF* estimation is used for depth superresolution of freely moving objects in a scene. Therefore, we compute the *SF* for consecutive depth and intensity image pairs in a sequence of T frames. The point set of each acquisition is then back propagated into the first frame solely through the *SF* vectors at each point by $\mathbf{X}_1(t) = \mathbf{X}_t - \sum_{i=1}^{t} \mathbf{u}_i, \forall t = T...1$. The superresolved depth image results by back-projecting all point sets $\mathbf{X}_1(t)$ into a higher resolution image space Ω_H. If multiple 3D points map in the same image pixel, the median depth of these points is taken. For this experiment we use $T = 10$ consecutive images from the real world scene shown in Fig. 6. The resulting depth map has $2.5\times$ the size of the original input images. The visual results for a rigid and a non-rigid movements of our superresolution approach compared to the first input depth map are shown in Fig. 5. The sequence and intermediate results are depicted in the supplementary video.



(a) Low resolution input (b) Superresolved output

Figure 5. Evaluation of depth image superresolution from *SF* on real image sequences. In (a) the object snippet of the first input depth map is shown. In (b) the corresponding superresolution result is shown with a lateral resolution of $2.5\times$ the input size.

This experiment shows the applicability of our *SF* method to the problem of depth image superresolution. Enabled by the accuracy of our method the superresolution can be calculated by a simple propagation of 3D measurements by the *SF*. This superresolution through *SF* delivers very sharp results even for non-rigidly moving objects.

6. Conclusion

We propose a method for the estimation of *SF* from depth and intensity data. The estimation is formulated as a convex energy minimization problem. We directly utilize the depth information through projection and warping in 3D space to formulate both a depth and an intensity image constraint. The regularization is formulated as a higher order penalizer to cope with smooth flow transitions, which occur at rotations or non-rigid movements, while sharp boundaries of the flow field are preserved. We further use the input depth image to weight and direct this regularization by an anisotropic diffusion tensor. In a quantitative and qualitative evaluation we show that our method clearly outperforms existing *SOTA* methods for *SF* and *OF* calculation in terms of speed and accuracy. We further give examples for the usage of *SF* applied to various computer vision problems. We show the applicability for depth image superresolution and camera pose estimation which are enabled by the accuracy of our method. As a future perspective, we think of combining the properties of camera and object pose estimation together with depth superresolution in a framework for dense real-time mapping of arbitrary non-rigid scenes without an explicit model of the scene.

Acknowledgments

This work was supported by *Infineon Technologies Austria AG* and the Austrian Research Promotion Agency (FFG) under the *FIT-IT Bridge* program, project #838513 (TOFUSION).

References

[1] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. In *CVPR*, 2010. 1, 2, 5, 6

[2] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journ. on Imaging Sciences*, 3(3):492– 526, 2010. 3

[3] J. Cech, J. Sanchez-Riera, and R. Horaud. Scene flow estimation by growing correspondence seeds. In *CVPR*, 2011. 2

[4] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journ. of Mathematical Imaging and Vision*, 40:120 –145, 2011. 4

[5] N. Cornelius and T. Kanade. Adapting optical-flow to measure object motion in reflectance and x-ray image sequences). 1984. 3

[6] X. R. Evan Herbst and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *ICRA*, 2013. 2

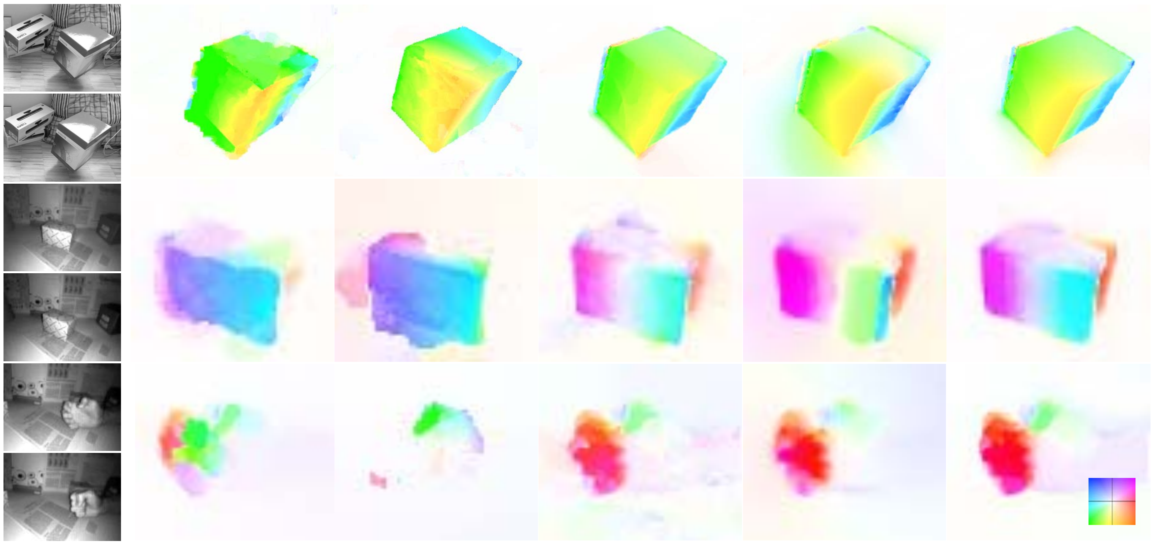

(a) intensity (b) *NL-TV-NCC* [29] (c) *SphereFlow* [12] (d) *aTGV-SF TV-reg* (e) *aTGV-SF w/o tensor* (f) *aTGV-SF*

Figure 6. Evaluation of our *SR* method on real image sequences. In (a) the input intensity images are shown. In the first and second row of (b-e) the flow results for a rotation/translation of a box (rigid movement) acquired wit a K4Wv2 and PMD Nano camera are shown. In the third row of (b-e) the flow result for a closing of a hand (non-rigid movement) is shown. Each scene is evaluated for *NL-NCC OF* (b), SphereFlow of [12] (c), *aTGV-SF* without a second order regularization (d), *aTGV-SF* without anisotropic diffusion (e) and our full method (f). The motion key is shown in the bottom right of (f).

[7] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *ICCV*, December 2013. 4

[8] J.-M. Gottfried, J. Fehr, and C. Garbe. Computing range flow from multi-modal kinect data. In *ISVC*, 2011. 2

[9] S. Hadfield and R. Bowden. Kinecting the dots: Particle based scene flow from depth sensors. In *ICCV*, 2011. 2, 5

[10] S. Hadfield and R. Bowden. Scene particles: Unregularized particle-based scene flow estimation. *TPAMI*, 36(3):564–576, 2014. 6

[11] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(13):185 –203, 1981. 1, 2

[12] M. Hornáček, A. Fitzgibbon, and C. Rother. Sphereflow: 6dof scene flow from rgb-d pairs. In *CVPR*, 2014. 2, 5, 6, 8

[13] P. J. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *Ann. Stat.*, 1(5):799 –821, 1973. 4

[14] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 2, 5, 6

[15] A. Letouzey, B. Petit, and E. Boyer. Scene flow from depth and color images. In *BMVC*, 2011. 2

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conference on Artificial Intelligence*, 1981. 2

[17] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *TPAMI*, 8(5):565 –593, 1986. 4

[18] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 6

[19] PMD Technologies. Siegen, Germany. *Camboard Nano*. 4

[20] J. Quiroga, F. Devernay, and J. Crowley. Scene flow by tracking in intensity and depth data. In *CVPR Workshops*, 2012. 2, 5

[21] J. Quiroga, F. Devernay, and J. L. Crowley. Local/global scene flow estimation. In *ICIP*, 2013. 2, 5, 6

[22] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *ECCV*. 2010. 2

[23] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *IEEE Intelligent Vehicles Symposium*, 2012. 4

[24] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003. 5

[25] D. Sun, S. Roth, and M. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, pages 1–23, 2013. 2, 5, 6

[26] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV*, 1999. 1, 2

[27] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *ICCV*, 2013. 2

[28] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3d motion understanding. *IJCV*, 95(1):29–51, 2011. 1, 2

[29] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *CVPR*, 2010. 5, 6, 8

[30] X. Zhang, D. Chen, Z. Yuan, and N. Zheng. Dense scene flow based on depth and multi-channel bilateral filter. In *ACCV*, 2012. 2, 5