Int J Comput Vis
DOI 10.1007/s11263-012-0542-7

Multi-view Scene Flow Estimation: A View Centered Variational Approach

Tali Basha · Yael Moses · Nahum Kiryati

Received: 7 September 2010 / Accepted: 23 May 2012

Abstract We present a novel method for recovering the 3D structure and scene flow from calibrated multi-view sequences. We propose a 3D point cloud parametrization of the 3D structure and scene flow that allows us to directly estimate the desired unknowns. A unified global energy functional is proposed to incorporate the information from the available sequences and simultaneously recover both depth and scene flow. The functional enforces multi-view geometric consistency and imposes brightness constancy and piecewise smoothness assumptions directly on the 3D unknowns. It inherently handles the challenges of discontinuities, occlusions, and large displacements. The main contribution of this work is the fusion of a 3D representation and an advanced variational framework that directly uses the available multi-view information. This formulation allows us to advantageously bind the 3D unknowns in time and space. Different from optical flow and disparity, the proposed method results in a nonlinear mapping between the images' coordinates, thus giving rise to additional challenges in the optimization process. Our experiments on real and synthetic data demonstrate that the proposed method successfully recovers the 3D structure and scene flow despite the complicated nonconvex optimization problem.

Keywords 3D structure · Scene flow · Multiple view

An earlier version of part of this work appear in CVPR 2010 (Basha et al. 2010).

T. Basha (✉) · N. Kiryati
School of Electrical Engineering, Tel Aviv University, Tel Aviv, 69978, Israel
e-mail: talib@eng.tau.ac.il

Y. Moses
Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya 46150, Israel

1 Introduction

The structure and motion of objects in a 3D space is an important characteristic of dynamic scenes. Reliable 3D motion maps can be utilized in many applications, such as surveillance, tracking, dynamic 3D scene analysis, autonomous robot navigation, 3D display devices, or virtual reality. In the last decade, an emerging field of research has addressed the problem of *scene flow* computation. Scene flow is defined as a dense 3D motion field of a nonrigid 3D scene (Vedula et al. 1999). It follows directly from this definition that 3D surface recovery must be an essential part of any scene flow algorithm, unless it is given *a priori*. Our objective is to simultaneously compute the 3D structure and scene flow from a multi-camera system. The system consists of N calibrated and synchronized cameras with overlapping fields of view. A unified variational framework is proposed to incorporate the information from the available sequences and simultaneously recover both depth and scene flow. To describe our method, we next elaborate on the parametrization of the problem, the integration of the spatial and temporal information from the set of sequences, and the setting of a global energy functional together with the variational framework used for solving it.

Most existing methods for scene flow and surface estimation parameterize the problem in 2D rather than 3D (e.g., Zhang and Kambhamettu 2000, 2001; Vedula et al. 2005; Isard and MacCormick 2006; Min and Sohn 2006; Huguet and Devernay 2007; Wedel et al. 2008; Li and Sclaroff 2008; Pock et al. 2008). That is, they compute disparity (stereo), which is the projection of the desired 3D shape, and the optical flow, which is the projection of the 3D motion (Fig. 1b). The relation between the scene flow and its projection is presented in Fig. 1a. Assuming that reliable and consistent solutions of both stereo and optical flow are given, the scene

Published online: 14 June 2012

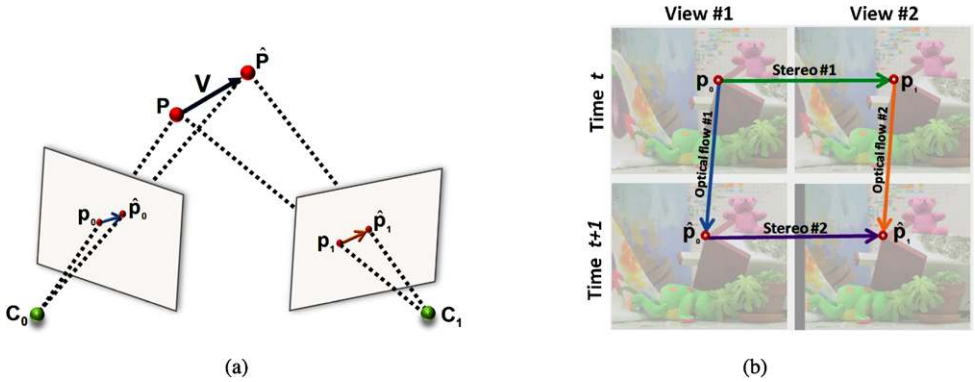

Fig. 1 (**a**) The point **P** is projected to pixels $\mathbf{p}_0$ and $\mathbf{p}_1$ on cameras C_0 and C_1, respectively. The new 3D location at $t+1$ is given by $\widehat{\mathbf{P}} = \mathbf{P} + \mathbf{V}$ and it is projected to $\widehat{\mathbf{p}}_0$ and $\widehat{\mathbf{p}}_1$. In this example, **V** is the 3D motion while $\mathbf{v} = \hat{p}_0 - p_0$ is the optical flow. (**b**) The 2D relation between corresponding points in two views at two time steps (optical flow and disparity fields)

flow and the 3D structure can be directly computed. This can be done, for example, by obtaining the 3D shape from the stereo in two time steps and deriving the scene flow from the optical flow in one of the images.

We propose a 3D point cloud parametrization of the 3D structure and 3D motion, with respect to a reference view (often referred as 2.5D parameterization). That is, for each pixel in a reference view, a depth value, **P**, and a 3D motion vector, **V**, are computed (see Fig. 1a). A similar parametrization for only 3D reconstruction was used by Robert and Deriche (1996). The advantage of using 3D rather than 2D parametrization is that it allows primary assumptions to be imposed on the unknowns prior to their projection. For example, a constant 3D motion field of a scene may project to a discontinuous 2D field (Fig. 2). Hence, in this example, smoothness assumptions hold for 3D parametrization but not for 2D. In addition, 3D parametrization allows direct extension to multiple views, without changing the problem's dimension. That is, the number of unknowns remains minimal, regardless of the number of views. This is in contrast to 2D parameterization where each additional view requires an additional set of parameters (e.g., disparity or optical flow maps).

We suggest coupling the spatio-temporal information by simultaneously recovering the scene flow and 3D structure. This approach is in contrast with previous approaches that decouple the scene flow and 3D structure problems (e.g., Vedula et al. 1999, 2005; Zhang and Kambhamettu 2000, 2001; Carceroni and Kutulakos 2002; Pons et al. 2007; Wedel et al. 2008). When the scene flow and 3D structure are decoupled, the two problems are solved sequentially. As a result, the spatio-temporal information is not fully utilized. In Vedula et al. (2005), for example, the optical flow field is computed independently for each camera without imposing consistency between the flow fields. Another example of the limitations of decoupling is the study by Wedel et al. (2008), where consistency is enforced on the stereo and motion solutions. Since the disparity map is computed separately, the results are still sensitive to its errors. Previous approaches for simultaneous recovery of scene flow and 3D structure help overcome these limitations (e.g., Vedula et al. 2000; Isard and MacCormick 2006; Min and Sohn 2006; Huguet and Devernay 2007; Neumann and Aloimonos 2002) but most rely on and hence suffer from the limitations of 2D parametrization; in particular, they are limited to two views. Our method simultaneously utilizes the multi-view information using 3D representation to improve the stability of the results and reduce possible ambiguities. (We extend on other methods that couple the multi-view information using 3D representation in Sect. 1.1).

The 3D parametrization and the spatio-temporal information from the set of sequences are used to define a global energy functional. The energy functional incorporates the multi-view geometry with a brightness constancy (BC) assumption (data term). Regularization is imposed by assuming piecewise smoothness directly on the 3D motion and depth. We avoid the linearization of the data term constraints to allow large displacements between frames. Moreover, discontinuities in both 3D motion and depth are preserved by using nonquadratic cost functions. This approach is motivated by the state-of-the-art optical flow variational approach of Brox et al. (2004). Our method is the first to extend it to multiple views and 3D parametrization. The minimization of the resulting nonconvex functional is obtained by solving the associated Euler-Lagrange equations. We follow a multi-resolution approach coupled with an image-warping strategy.

We tested our method on challenging real and synthetic data. When ground truth is available, we propose a new evaluation of scene flow based on the 3D errors rather than the conventional 2D error. We argue that the 2D errors traditionally used for evaluating stereo and optical flow algorithms do not necessarily correlate with the 3D errors. In particular, we show that the ranking of stereo algorithms (e.g., Scharstein and Szeliski 2002) may change when the 3D errors are considered.

The main contribution of this paper is to combine a novel 3D formulation with an accurate global energy functional that explicitly describes the desired assumptions on the 3D structure and scene flow. The functional inherently handles

the challenges of discontinuities, occlusions, and large displacements. Combining our 3D representation in that variational framework leads to a better constraint problem that directly utilizes the information from multi-view sequences. As demonstrated in our experiments, we successfully recover the 3D structure and scene flow despite the challenging nonconvex optimization problem.

The rest of the paper is organized as follows. We begin with reviewing related studies in Sect. 1.1. Section 2 describes our method. Section 3 provides insight into our quantitative 3D evaluation measures. In Sect. 4 we present the experimental results. We conclude in Sect. 5.

1.1 Related Work

To the best of our knowledge, our view-centered 3D point cloud representation has not been previously considered for the scene flow recovery problem. Other 3D parameterizations, that are not view dependent, were studied: 3D array of voxels, Vedula et al. (1999), various mesh representations (Furukawa and Ponce 2008; Courchay et al. 2009; Neumann and Aloimonos 2002) and dynamic surfels (Carceroni and Kutulakos 2002). In contrast to our method, each of these 3D representations can provide a complete, view-independent 3D description of the scene. However, the scene that can be considered is often limited by the representation (e.g., a single moving object) and a large number of cameras is required in order to benefit from their choice of parametrization. In addition, in these representations, the discretization of the 3D space is often independent of the actual 2D resolution of the available information from the images.

The studies most closely related to ours in the sense of numeric similarity are (Huguet and Devernay 2007; Wedel et al. 2008). Huguet and Devernay (2007) proposed to simultaneously compute the optical flow field and two disparity maps (in successive time steps), while Wedel et al. (2008) decoupled the disparity at the first time step from the rest of the computation. Both extend the variational framework of Wedel et al. (2008) for solving for scene flow and structure estimation. In these studies regularization is imposed on the disparity and optical flow (2D formulation), while our assumptions refer directly to the 3D unknowns. Nor were these methods extended to multiple views.

A multi-view energy minimization framework was presented by Zhang and Kambhamettu (2000). A hierarchical rule-based stereo algorithm was used for initialization. Their method imposed optical flow and stereo constraints while preserving discontinuities using image segmentation information. Each view results in an additional set of unknowns, and the setup is restricted to a parallel camera array. Another multi-view method was suggested by Pons et al. (2007). They use a 3D variational formulation in which the prediction error of the shape and motion is minimized by using a level-set framework. However, the shape and motion are sequentially computed.

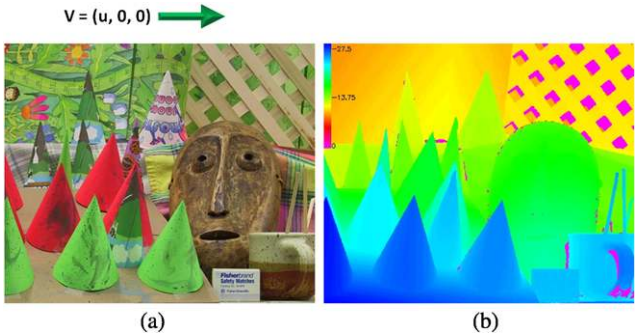

Fig. 2 (**a**) The Middlebury stereo dataset, *Cones*. The scene flow, **V**, is constant for all image points since only the camera translates. (**b**) The horizontal component of the projected scene flow, the optical flow. It depends on the 3D point location

There are only a few multi-view methods that use 3D representations and simultaneously solve the 3D surface and motion. Neumann and Aloimonos (2002) modeled the object by a time-varying subdivision hierarchy of triangle meshes, optimizing the position of its control points. However, their method was applied only to scenes which consist of one connected object. Furukawa and Ponce (2008) constructed an initial polyhedral mesh at the first frame. It is tracked assuming locally rigid motion and globally non-rigid deformation. Courchay et al. (2009) represented the 3D shape as an animated mesh. The shape and motion are recovered by optimizing the positions of its vertices under the assumption of photo-consistency and smoothness of both the surface and 3D motion. Nevertheless, both methods Courchay et al. (2009) and Furukawa and Ponce (2008) are limited due to the fixed mesh topology.

2 The Method

Our goal is to simultaneously reconstruct the 3D surface of a 3D scene and its scene flow (3D motion) from N static cameras. The cameras are assumed to be calibrated and synchronized, each providing a pair of successive frames of the scene. We assume brightness constancy (BC) in both spatial (different viewpoints) and temporal (3D motion) domains. We formulate an energy functional which we minimize in a variational framework by solving the associated Euler-Lagrange equations.

2.1 System Parameters and Notations

Consider a set of N calibrated and synchronized cameras, $\{C_i\}_{i=0}^{N-1}$. Let $I_i(x, y, t) : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$, be the sequence taken by camera C_i. Let M^i be the 3×4 projection matrix of camera C_i. The projection of a 3D surface point

$\mathbf{P} = (X, Y, Z)^T$ onto an image of the ith sequence at time t is given by:

$$\mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \frac{[M^i]_{1,2}[\mathbf{P}\ 1]^T}{[M^i]_3[\mathbf{P}\ 1]^T}, \tag{1}$$

where $[M^i]_{1,2}$ is the 2×4 matrix which contains the first two rows of M^i and $[M^i]_3$ is the third row of M^i.

Let $\mathbf{V} = (u, v, w)^T$ be the 3D displacement vector of the 3D point $\mathbf{P}$ (in our notation bold characters represent vectors). The new location of a 3D point $\mathbf{P}$ after the displacement $\mathbf{V}$ is denoted by $\widehat{\mathbf{P}} = \mathbf{P} + \mathbf{V}$. Its projection onto the ith image at time $t + 1$ is denoted by $\widehat{\mathbf{p}}_i$ (see Fig. 1a).

Assume without loss of generality that the 3D points are given in the coordinate system of the reference camera, C_0. In this case, the X and Y coordinates are functions of Z and are given by the back projection:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = Z \begin{pmatrix} x/s_x \\ y/s_y \end{pmatrix} - Z \begin{pmatrix} o_x/s_x \\ o_y/s_y \end{pmatrix}, \tag{2}$$

where s_x and s_y are the scaled focal lengths, (o_x, o_y) is the principal point, and $(x, y)^T$ are the reference image coordinates. We directly parameterize the 3D surface and scene flow with respect to (x, y) and t. In particular, given the time step, t, the surface and scene flow are represented as the 3D functions, $\mathbf{P}(x, y, t) : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and $\mathbf{V}(x, y, t) : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$, respectively. That is,

$$\mathbf{P}(x, y, t) = \big(X(x, y, t),\ Y(x, y, t),\ Z(x, y, t)\big)^T, \tag{3}$$

$$\mathbf{V}(x, y, t) = \big(u(x, y, t),\ v(x, y, t),\ w(x, y, t)\big)^T. \tag{4}$$

Note that $\mathbf{P}(x, y, t + 1)$ is the 3D surface point which is projected to pixel $\mathbf{p} = (x, y)^T$ at time $t + 1$. Obviously, it is different from $\widehat{\mathbf{P}}$, which is projected to a different image pixel $\widehat{\mathbf{p}}$ (unless there is no motion). From this point, we will refer to $\mathbf{P}$ and $\mathbf{V}$ at a fixed time step, t. Hence, the temporal dependency of $\mathbf{P}$ and $\mathbf{V}$ will be disregarded.

For each image point in the reference camera, (x, y), and a single time step, there are six unknowns: three for $\mathbf{P}$ and three for $\mathbf{V}$. However, since X and Y can be determined by Eq. (2) as functions of Z and (x, y), there are only four unknowns for each image pixel. In particular, the 3D point $\mathbf{P}$ is given by:

$$\mathbf{P}(x, y) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = Z(x, y) \begin{pmatrix} x/s_x - o_x/s_x \\ y/s_y - o_y/s_y \\ 1 \end{pmatrix}. \tag{5}$$

We aim to recover Z and $\mathbf{V}$ as functions of (x, y), using the N pairs of images.

In this representation, the number of unknowns is independent of the number of cameras. Hence, a multi-view system can be efficiently used without changing the dimensions of the problem. This is in contrast to previous methods that use 2D parametrization, e.g., Huguet and Devernay (2007), Wedel et al. (2008), Li and Sclaroff (2008), Strecha et al. (2003), where additional cameras require additional sets of unknowns (e.g., optical flow or disparity field). Moreover, our representation does not require image rectification.

2.2 The Energy Functional

The total energy functional we aim to minimize is a sum of two terms:

$$E(Z, \mathbf{V}) = E_{Data} + \alpha E_{Smooth}. \tag{6}$$

The data term E_{data} expresses the fidelity of the result to the model. Recovering the surface and scene flow by the minimization of E_{data} alone is an ill-posed problem. Hence, regularization is used, mainly to deal with ambiguities (low texture regions) and image noise. In addition, the regularization is used to obtain solutions for occluded pixels (see Sect. 2.4). The relative impact of each of the terms is controlled by the regularization parameter $\alpha > 0$. Next, we elaborate on each of these terms.

Data Term The data term imposes the brightness constancy assumption in both spatial and temporal domains. That is, the intensity of a 3D point's projection onto different images before and after the 3D displacement does not change. Additionally, our 3D parametrization forces the solution to be consistent with the 3D geometry of the scene and the camera parameters. In particular, the epipolar constraints are satisfied.

The brightness constancy assumption is generalized for all N cameras and for both time steps. The data term is obtained by integrating the sum of three penalizers over the reference image domain. BC_m penalizes deviation from the brightness constancy assumption before and after 3D displacement; BC_{s_1} and BC_{s_2} penalize deviation from the brightness constancy assumption between the reference view and each of the other views at time t and $t + 1$, respectively. Formally the penalizers for each pixel are defined by:

$$\begin{aligned}
BC_m(Z, \mathbf{V}) &= \sum_{i=0}^{N-1} c_m^i \Psi\left(\left|I_i(\mathbf{p}_i, t) - I_i(\widehat{\mathbf{p}}_i, t + 1)\right|^2\right), \\
BC_{s_1}(Z) &= \sum_{i=1}^{N-1} c_{s_1}^i \Psi\left(\left|I_0(\mathbf{p}_0, t) - I_i(\mathbf{p}_i, t)\right|^2\right), \\
BC_{s_2}(Z, \mathbf{V}) &= \sum_{i=1}^{N-1} c_{s_2}^i \Psi\left(\left|I_0(\widehat{\mathbf{p}}_0, t + 1) - I_i(\widehat{\mathbf{p}}_i, t + 1)\right|^2\right),
\end{aligned} \tag{7}$$

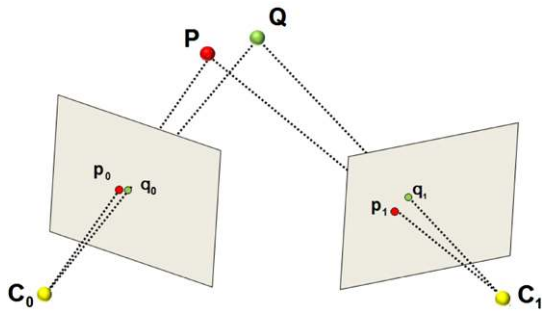

Fig. 3 Two adjacent pixels in camera C_0 correspond to two distance pixels in camera, C_1. Hence, the gradient of pixel p_0 in C_0 is generally different from the gradient of its corresponding pixel, p_1 in C_1

where $\Psi(s^2)$ is a chosen cost function and c_*^i is a binary mask that omits occluded pixels from the computation, since they are not expected to satisfy the brightness constancy assumption (see Sect. 2.4). We use a nonquadratic robust cost function $\Psi(s^2) = \sqrt{s^2 + \varepsilon^2}$ ($\varepsilon = 0.0001$), which is a smooth approximation of L_1 (see Brox et al. 2004), for reducing the influence of outliers on the functional. The outliers are pixels that do not comply with the model due to noise, lighting changes, reflections or occlusions. In this formulation, no linear approximations are made; hence large displacements between frames are allowed.

The parameterization used by our method (defined in Sect. 2.1) leads to nonlinear mappings between the images' coordinate systems to the reference image coordinate system. We extend on these mappings in Appendix A. Observe that when 2D parameterization is considered, namely optical flow and disparity, the mappings between the images' coordinates are given by adding to those coordinates the optical flow and/or the disparity fields to images' coordinates. This simple mapping is probably one of the reasons that 2D parameterization is often chosen to parametrize the scene flow.

In natural scenes the BC assumption does not necessarily hold for all pixels in all frames, in particular when considering wide baseline setup. To overcome this problem, previous studies for estimating optical flow (e.g., Brox et al. 2004) or scene flow (e.g., Huguet and Devernay 2007) imposed an additional gradient constancy assumption in order to allow deviation in the gray value. Nevertheless, since the gradient is viewpoint dependent (due to the foreshortening effect), this assumption does not hold in the spatial domain (see Fig. 3). Hence, we chose not to impose the additional gradient constancy assumption. The robustness of our method to deviation from the BC assumption is obtained by using multiple views. That is, since the data term is given by integrating the deviation from the BC assumption over all views, each view has a relative impact on the total deviation from the BC assumption.

Smoothness Term We assume that both the 3D motion field and surface are changing piecewise smoothly w.r.t. reference camera. Deviations from this model are usually penalized by using a total variation regularizer, which is generally the L_1 norm of the field derivatives. Here we use the same robust function $\Psi(s^2)$ for preserving discontinuities in both the scene flow and depth. Using the notation, $\nabla = (\partial_x, \partial_y)^T$, this can be expressed as:

$$\begin{aligned} S_m(\mathbf{V}) &= \Psi\left(\left|\nabla u(x,y)\right|^2 + \left|\nabla v(x,y)\right|^2 + \left|\nabla w(x,y)\right|^2\right), \\ S_s(Z) &= \Psi\left(\left|\nabla Z(x,y)\right|^2\right), \end{aligned} \tag{8}$$

where S_m and S_s are the penalizeres of deviation from the motion and shape (piecewise) smoothness assumption, respectively. Note that the first order regularizer gives priority to fronto-parallel solutions. In future work we intend to explore a general smoothness constraint that is unbiased to a particular direction. For example, a second order smoothness prior (Woodford et al. 2009) might be more suitable in our framework. In addition, the current regulazier depends on the depth range in the scene. Therefore, a normalization that takes into account the depth values in each pixel may be desirable.

The total energy function is obtained by integrating the penalty (Eqs. (7)–(8)) over all pixels in the reference camera, Ω:

$$E(Z,\mathbf{V}) = \int_\Omega \Big[\underbrace{BC_m + BC_s}_{Data} + \alpha \underbrace{(S_m + \mu S_s)}_{Smooth}\Big] dxdy, \tag{9}$$

where $BC_s = BC_{s_1} + BC_{s_2}$, and $\mu > 0$ is a parameter used to balance the motion and the surface smoothness.

2.3 Optimization

We wish to find the functions $Z, \mathbf{V}$ that minimize our functional (Eq. (9)) by means of calculus of variations. Calculus of variations supplies a necessary condition to achieve a minimum of a given functional, which is essentially the vanishing of its first variation. This leads to a set of partial differential equations (PDEs) called *Euler-Lagrange equations*. In our case the associated Euler-Lagrange equations can generally be written as:

$$\left(\frac{\partial E}{\partial Z}, \frac{\partial E}{\partial u}, \frac{\partial E}{\partial v}, \frac{\partial E}{\partial w}\right)^T = 0. \tag{10}$$

2.3.1 Euler-Lagrange Equations

Consider the points $\mathbf{P}$, $\widehat{\mathbf{P}}$, their sets of projected points $\{\mathbf{p}_i\}_{i=0}^{N-1}$, $\{\widehat{\mathbf{p}}_i\}_{i=0}^{N-1}$, and the sequences $\{I_i\}_{i=0}^{N-1}$. We use the following abbreviations for the difference in intensities between corresponding pixels in time and space:

$$\begin{aligned} \Delta_i &= I_i(\mathbf{p}_i, t) - I_0(\mathbf{p}_0, t), \\ \widehat{\Delta}_i &= I_i(\widehat{\mathbf{p}}_i, t+1) - I_0(\widehat{\mathbf{p}}_0, t+1), \\ \Delta_i^t &= I_i(\widehat{\mathbf{p}}_i, t+1) - I_i(\mathbf{p}_i, t). \end{aligned} \tag{11}$$

We use subscripts to denote the image derivatives. Using the aforementioned notations, the nonvanishing terms of the equations with respect to Z and u result in:

$$
\begin{aligned}
0 = & \sum_{i=0}^{N-1} \Psi'\left(\left(\Delta_i^t\right)^2\right)\Delta_i^t \cdot \left(\Delta_i^t\right)_Z \\
& + \sum_{i=1}^{N} \Psi'\left((\Delta_i)^2\right)\Delta_i \cdot (\Delta_i)_Z \\
& + \sum_{i=1}^{N-1} \Psi'\left((\widehat{\Delta}_i)^2\right)\widehat{\Delta}_i \cdot (\widehat{\Delta}_i)_Z \\
& - \alpha\mu \cdot \operatorname{div}\left(\Psi'\left(|\nabla Z|^2\right)\nabla Z\right), \qquad (12)
\end{aligned}
$$

$$
\begin{aligned}
0 = & \sum_{i=0}^{N-1} \Psi'\left(\left(\Delta_i^t\right)^2\right)\Delta_i^t \cdot \left(\Delta_i^t\right)_u \\
& + \sum_{i=1}^{N} \Psi'\left((\widehat{\Delta}_i)^2\right)\widehat{\Delta}_i \cdot (\widehat{\Delta}_i)_u \\
& - \alpha \cdot \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2 + |\nabla w|^2\right)\nabla u\right), \qquad (13)
\end{aligned}
$$

with the Neumann boundary condition: $\partial_n Z = \partial_n u = \partial_n v = \partial_n w = 0$, where n is the normal to the image boundary. The Euler-Lagrange equations with respect to v and w are similar to Eq. (13). Observe that the first variation of the functional with respect to Z involves computing the derivatives of all images (none of them vanish). This enforces the desired synergy of the data from all sequences.

At this point, it is worth noting that the pixels are nonlinear functions of the 3D unknowns due to perspective projection. As a result, the computation of image derivatives with respect to Z and $\mathbf{V}$ requires using the chain rule, often in a nontrivial manner (see Appendices A–B).

2.3.2 Numerics

Our parametrization and functional represent precisely the desired model (no approximations are made), resulting in a complicated minimization problem. In particular, the use of nonlinearized data terms and nonquadratic penalizers yields a nonlinear system in the four unknown functions Z and $\mathbf{V}$ (e.g., Eqs. (12)–(13)). Therefore, one has to deal with the problem of multiple local minima as a result of the nonconvex functional. In our method, the derivation and discretization of the equations results in additional complexity since the perspective projection is nonlinear in the unknowns Z and $\mathbf{V}$ (see Appendices A–B).

We cope with these challenges by using a multi-resolution warping method coupled with two nested fixed point iterations as previously suggested by Brox et al. (2004). The multi-resolution approach is employed by downsampling each input image to an image pyramid with a scale factor η. The original projection matrices are modified to suit each level by scaling the intrinsic parameters of the cameras. The amplitude of our 3D unknowns remains fixed regardless to the pyramid level used. (Note that the amplitude of the optical flow and the disparity is scaled according to the pyramid level.) Starting from the coarsest level, the solution is computed at each level and then utilized to initiate the lower (finer) level. This justifies the assumption of small changes in the solution between consecutive levels. Thus, the equations can be partially linearized by Taylor expansion. Furthermore, the effect of "smoothing" the functional in the "coarse to fine" approach increases the chance of converging to the global minimum. We wish to avoid oversmoothing at the low resolution levels by keeping the relative impact of the smoothness term the same in all levels. This is obtained by scaling the smoothness term $\alpha_\ell = \alpha \cdot \eta^\ell$ with respect to the pyramid level, ℓ. The required resolution of the coarsest level depends on the quality of the initial depth or flow maps. However, if the resolution is too low, small objects might be oversmoothed. We discuss this issue in Sect. 4.

The solution in a given pyramid level is obtained from two nested fixed point iterations that are responsible for removing the nonlinearity in the equations. The outer iteration is responsible for the linearizion of the expressions given in Eq. (11) using the first order Taylor expansion. At each outer iteration, k, small increments in the solutions, dZ^k and $d\mathbf{V}^k = (du^k, dv^k, dw^k)^T$, are estimated. Next, the total solution is updated using $Z^{k+1} = Z^k + dZ^k$ and $\mathbf{V}^{k+1} = \mathbf{V}^k + \mathbf{dV}^k$, the images are rewarped accordingly, and the image derivatives are recomputed (see Appendix B). The inner loop is responsible for removing the nonlinearity that resulted from the use of the function Ψ. At each inner iteration a final linear system of equations is obtained by keeping Ψ' expressions fixed (see Appendix C). The final linear system is solved by applying the *successive overrelaxation* (SOR) method (Young 1954).

2.4 Occlusions

Scene points viewed by the reference camera at time t, may be occluded in one or more of the other images, taken from a different viewpoint or at different time steps. Our method defines the correspondence between pixels in two images using the projection of a 3D point to each of the images. Hence, when a point is occluded in one image, its computed correspondence is incorrect. In particular, the brightness constancy assumption is not satisfied in this case.

The use of a multi-view system in our method increases the chances of a point to be occluded in at least one of the images, especially those taken from different distant viewpoints. Therefore, the occluded pixels cannot be negligible or treated as outliers. To overcome this problem, the associated component of occluded pixels should be omitted from

Algorithm 1 Calculate occlusion map, c^i_{s1}: zero value for occluded

$S \leftarrow 0$ {S—source map of reference image coordinates}
$c^i_{s1} \leftarrow 1$ {c^i_{s1}—occlusion map}
{Going over all 3D points w.r.t. the reference view}
for each 3D point $\mathbf{P}(\mathbf{p}_0)$ **do**
  $\mathbf{p}_i = proj(M^i, \mathbf{P}(\mathbf{p}_0))$ {M^i—the projection matrix}
  if $S(\mathbf{p}_i) == 0$ **then**
    $S(\mathbf{p}_i) = \mathbf{p}_0$ {$\mathbf{p}_i$ is approached for the first time}
  else
    $\mathbf{q} = S(\mathbf{p}_i)$
    if $\|\mathbf{P}(\mathbf{p}_0) - \mathbf{P}(\mathbf{q})\| > th$ **then**
      {p_0 and q are different 3D points}
      if $\|\mathbf{P}(\mathbf{p}_0) - COP_i\| < \|\mathbf{P}(\mathbf{q}) - COP_i\|$ **then**
        $S(\mathbf{p}_i) = \mathbf{p}_0$ {saving closest point origin}
        $c^i_{s1}(\mathbf{q}) = 0$ {$\mathbf{p}_i$ is the occluded}
      else
        $c^i_{s1}(\mathbf{p}_0) = 0$ {$\mathbf{p}_i$ is occluded}
      end if
    end if
  end if
end for

the relevant data term. This is accomplished by computing for each view (other than the reference) three occlusion maps, c^i_*. Each map corresponds to the relevant penalizer in the data term (Eq. (7)). The computed maps are used as 2D binary masks on each of the data term components. Since we use multiple views, each scene point viewed by the reference camera is expected to be visible in at least one more view. If a point is visible only in the reference view, its solution would be determined by the smoothness term.

It is important to consider how the occluded pixels are determined. One approach can be to directly consider it as a part of the minimization problem (e.g., Ben-Ari and Sochen 2007; Ayvaci et al. 2010). For example, for computing the occluded pixels in optical flow, the optimization may include searching for a minimal sparse set of pixels that do not satisfy the brightness constancy assumption (Ayvaci et al. 2010). However, such methods do not take into account the scene geometry. When scene geometry and the camera parameters are known, the occluded pixels are uniquely determined; hence, the occluded pixels cannot be added as an additional set of unknowns.

We use the computed 3D shape and motion in a given iteration for determining the visibility of each 3D surface point in each of the cameras at each time step. (A similar approach was used in Huguet and Devernay (2007), Wedel et al. (2011).) A modified Z-buffering is applied for directly computing the occlusion maps. These maps are computed w.r.t. the reference image. For example, the map c^i_{s1} is computed by testing, for each pixel from camera i at time t, its origin in the reference image. When two pixels from the reference image are mapped to the same pixel in frame i, one of them is occluded. The occluded pixel is determined by the distances between the associated 3D points from the center of projection of camera i. The pseudocode of this algorithm is given in Algorithm 1. The other occlusion maps are computed in a similar manner. The maps are updated at each outer iteration in order to include the increments of the unknowns in the computation.

3 A Note on Error Evaluation

We evaluate scene flow in 3D rather than in 2D. That is, the error is defined by the deviation of the estimated 3D point, $\mathbf{P}(x, y)$, and 3D motion, $\mathbf{V}(\mathbf{x}, \mathbf{y})$, from their corresponding ground truth values, $\mathbf{P}_o(x, y)$ and $\mathbf{V}_o(\mathbf{x}, \mathbf{y})$. Various statistics over these errors can then be chosen. We use the *normalized root mean square* (*NRMS*) error, which is the percentage of the *RMS* error from the range of the observed values. The $NRMS_{\boldsymbol{P}}$ is defined by:

$$NRMS_{\boldsymbol{P}} = \frac{\sqrt{\frac{1}{N}\sum_{\Omega} \|\mathbf{P}(x,y)^T - \mathbf{P}_o(x,y)^T\|^2}}{\max(\|\mathbf{P}_o(x,y)\|) - \min(\|\mathbf{P}_o(x,y)\|)}, \tag{14}$$

where Ω denotes the integration domain (e.g., nonoccluded areas) and N is the number of pixels. Note that this measure is independent of the units of Z. Similarly, the $NRMS_{\boldsymbol{V}}$ error is computed for the 3D motion vector $\mathbf{V}$. In addition, the scene flow angular error is evaluated by computing the *absolute angular error* (*AAE*), for the vector $\mathbf{V}$.

Conventionally, evaluations of stereo, optical flow, and scene flow algorithms are performed in the image plane. That is, the computed error is the deviation of the projection of the erroneous values in 3D from their 2D ground truth (error of the disparity or the optical flow). The proposed 3D evaluation is motivated by the observation that the errors in 2D (in the image plane) do not necessarily approximate well nor correlate with the errors in 3D. In particular, the 2D error at a given pixel depends not only on the magnitude of the 3D error but also on the 3D error sign (toward or away from the camera). A simple example in Fig. 4 demonstrates how the sign of the 3D error affects the size of the 2D error. Furthermore, the 3D errors strongly depend on the depth of the point, $Z(x, y)$, as well as on the location within the image, (x, y). In particular, using Eq. (2) it can be shown that Eq. (14) can be written as:

$$NRMSE_{3D} = \frac{\sqrt{\frac{1}{N}\sum_{\Omega}(Z(x,y) - Z_o(x,y))^2 \cdot w(x,y)}}{\max(|Z(x,y)\cdot\sqrt{w(x,y)}|) - \min(|Z(x,y)\cdot\sqrt{w(x,y)}|)}, \tag{15}$$

where

$$w(x, y) = \left(\frac{x - o_x}{s_x}\right)^2 + \left(\frac{y - o_y}{s_y}\right)^2 + 1. \tag{16}$$

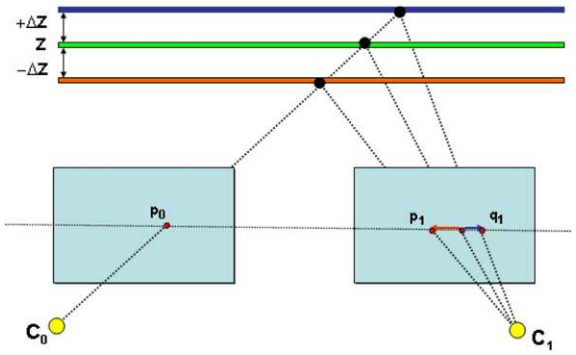

Fig. 4 Illustration of the difference between the disparity errors originating from the opposite sign of the 3D error: the *green line* represents the ground truth depth Z; the *blue* and *orange lines* represent positive and negative erroneous depths, respectively; and $\mathbf{p}_1$ and $\mathbf{q}_2$ are their projection onto camera C_1, which results in different absolute values of the disparity error

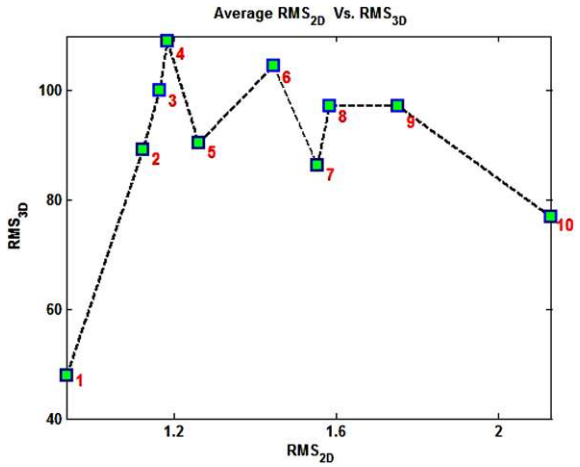

Fig. 5 Comparison between the ranking order in 2D and 3D: the computed average RMS error of each algorithm (numbered from one to ten) in 2D vs. 3D. The resulting nonmonotonic graph demonstrates the changes in the ranking

The 3D error's dependence on $w(x, y)$ and on $Z(x, y)$ is not taken into account by 2D error evaluation. Hence the correlation between the two types of evaluation can be weak. Thus, when comparing the results of 3D reconstruction or scene flow algorithms, the 3D evaluation may result in different ranking than the 2D.

To practically test this observation, we evaluated the results of the top ten ranked stereo algorithms in the Middlebury stereo evaluation (Scharstein and Szeliski) using 2D and 3D errors. We chose to compare the ranking using the RMS measure (since it does not require any error tolerance setting).

The errors were computed for three of the Middlebury stereo datasets, *Cones*, *Teddy* and *Venus* (Scharstein and Szeliski 2003), over three domains: all pixels, nonoccluded regions, and only discontinuous regions. The intrinsic parameters of the camera were set as explain in Sect. 4.1. Figure 5 shows, for each algorithm, the average RMS error (over the three datasets and the three domains) in 2D, versus the average RMS error in 3D. As expected, the results demonstrate that changes in the ranking indeed occur when RMS is considered. For example, the second and third ranked algorithm in 3D RMS are ranked as the tenth and the seventh in 2D RMS.

4 Experimental Results

Our algorithm was implemented in *C* using the *OpenCV* library.[1] Like all variational methods, our method requires initial depth and 3D motion maps. In general, any stereo algorithm can be used for obtaining an initial depth map. In all experiments the 3D motion field was simply initiated to zero. It is possible to improve the trivial motion initialization, by first fixing the initial depth at the coarser levels of the pyramid and optimizing only for U, V and W. The full optimization on both flow and depth can then start only from an intermediate level. However, the results next presented were obtained directly with the trivial motion initialization.

In the first two experiments, where the input images were rectified, we used the stereo algorithm proposed in Felzenszwalb and Huttenlocher (2006). In the third experiment, to avoid rectification of the input images, we used a naive initialization of two parallel planes. This initialization is very far from the real depth and scene flow, but as shown, is sufficient to converge to the correct solution. Clearly, using a more sophisticated initialization can improve the convergence time.

The running time of our method is the same order of magnitude as that of Huguet and Devernay (2007). In addition, one should consider the running time of the chosen stereo algorithm used for initialization. The code can probably be significantly accelerated by implementation on parallel architecture (e.g., GPU), however, it is not the focus of our method and is left for future research. We next elaborate on each of the experiments.

4.1 Egomotion Using Stereo Datasets

This experiment consists of a real 3D rigid translating scene viewed by two, three and four cameras. This scenario can also be regarded as a static scene viewed by a translating "camera array" where our method computes the egomotion of the cameras. The Middlebury stereo datasets, *Cones*, *Teddy* and *Venus* (Scharstein and Szeliski 2003), were used for generating the data (as in Huguet and Devernay 2007).

[1] The source code is publicly available.

Each dataset consists of 9 rectified images taken from equally spaced viewpoints. Eight of the images were considered as taken by four cameras at two time steps. Due to the camera setup, both the 2D and the 3D motion are purely horizontal. Still, while the 3D motion is constant over the entire scene, the 2D motion is generally different for each pixel. We do not make use of this knowledge when testing our algorithm (see Fig. 2).

Our method requires full calibration, which, however, is not available for these datasets. We set the projection matrix of each camera up to two degrees of freedom by using the known relative cameras' positions. One of the cameras is taken as the reference camera. Accordingly, the others cameras' extrinsic parameters are set as only translation along the horizontal axis with respect to the reference camera. The cameras' intrinsic parameters are computed by defining the viewing angle (chosen to be 30°), and the scaled focal lengths are uniquely determined by the image size. Note that this arbitrary choice of parameters may impair the quality of our results.

For comparison with the results of the scene flow algorithm proposed by Huguet and Devernay (2007), we project our results for $\mathbf{V}$ and Z onto the images. To evaluate the results, we compute the absolute angular error (*AAE*) for the optical flow and the root mean square error (*RMS*) for the optical flow and each of the disparity fields at time t and time $t+1$. These measurements are given in Table 1. We achieved significantly better results for the optical flow and disparity at time $t+1$ and similar results for the disparity at time t. There is an improvement of 46 %–54 % in the *RMS* error of the optical flow and 28 %–58 % in the *RMS* error of the disparity $t+1$. Furthermore, the advantage of using more than two views is demonstrated. As expected, the use of more than two views leads to better results for all the unknowns.

Table 1 The evaluated errors (w.r.t. the ground truth) of the projection of our scene flow and structure compared with the 2D results of Huguet and Devernay (2007). RMS error in the optical flow (O.F.), disparity at time t, and the disparity at time $t+1$. Also shown is the absolute angular error (AAE) corresponding to the optical flow.

		RMS			*AAE*
		O.F.	Disp. at t	Disp. at $t+1$	(deg)
Cones	4 Views	0.25	2.36	2.36	0.12
	2 Views	0.58	2.48	2.49	0.39
	Huguet and Devernay 2007	1.1	2.11	5.24	0.69
Teddy	4 Views	0.51	2.47	2.47	0.22
	2 Views	0.57	2.83	2.86	1.01
	Huguet and Devernay 2007	1.25	2.27	6.93	0.51
Venus	4 Views	0.13	0.9	0.9	1.09
	2 Views	0.16	1.06	1.06	1.58
	Huguet and Devernay 2007	0.31	0.97	1.48	0.98

4.2 Synthetic Data

4.2.1 Multi-View Rotating Sphere

We tested our method on a challenging synthetic scene viewed by five calibrated cameras. This sequence was generated in OpenGL and consists of a rotating sphere placed in front of a rotating plane. The plane is placed at $Z = 700$ (the units are arbitrary) and the center of the sphere at $Z = 500$ with radius of 200. Both plane and sphere are rotated, each around different 3D axes with different angles (see Fig. 6). Therefore, occlusions and large discontinuities in both motion and depth must be dealt with. The accuracy of the computed depth and 3D motion is demonstrated in Fig. 7 by comparing them with the ground truth. The results are quantitatively evaluated by computing the $NRMS_{\mathbf{P}}$, $NRMS_{\mathbf{V}}$ errors and the $AAE_{\mathbf{V}}$ (defined in Sect. 3). Table 2 summarizes the computed errors over three domains: all pixels, nonoccluded regions, and only continuous regions (namely, removing regions corresponding to discontinuities of the surface). An analysis of our results clearly shows that oversmoothing in the discontinuous areas accounts for most of the errors.

4.2.2 Orthographic Rotating Sphere

We tested our method on the *Rotating Sphere* dataset from Huguet and Devernay (2007). The scene represents a rotating textured sphere, where its two hemispheres rotate separately in opposite directions (see Fig. 8). The input images were generated as taken under orthographic projection, by two cameras related by rotation. However, our method assumes a perspective camera model. Hence, we interpreted the input images as taken by a parallel pair of cameras under perspective projection. The parameters of the cameras were chosen arbitrarily to be $s_x = s_y = 200$ (scaled focal length) and $T = 20$ (baseline). Such interpretation results in a different 3D scene (a distorted ball) and 3D motion; this is illustrated in longitudinal sections in Fig. 9. The background, where the initial disparity was set to zero, was treated as an occluded region in our implementation (since otherwise the depth would need to be set to infinity).

Figure 9 shows the recovered depth compared to the ground truth of the object along longitudinal sections. The recovered depth is almost perfect except in regions which have very large depth change (close to the boundaries). These significantly large gradients in depth are due to

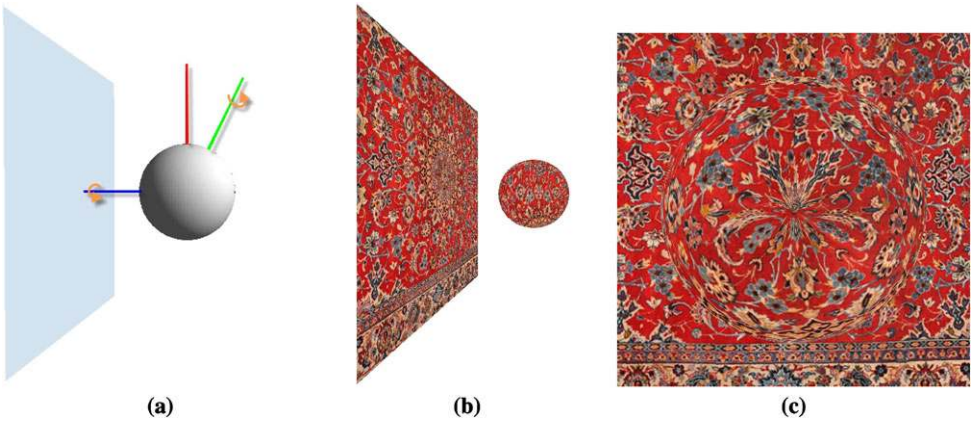

Fig. 6 (**a**) Illustration of the rotation axes. The sphere is rotating around the green axis and the plane around the blue one. (**b**) With texture. (**c**) The reference view before rotation

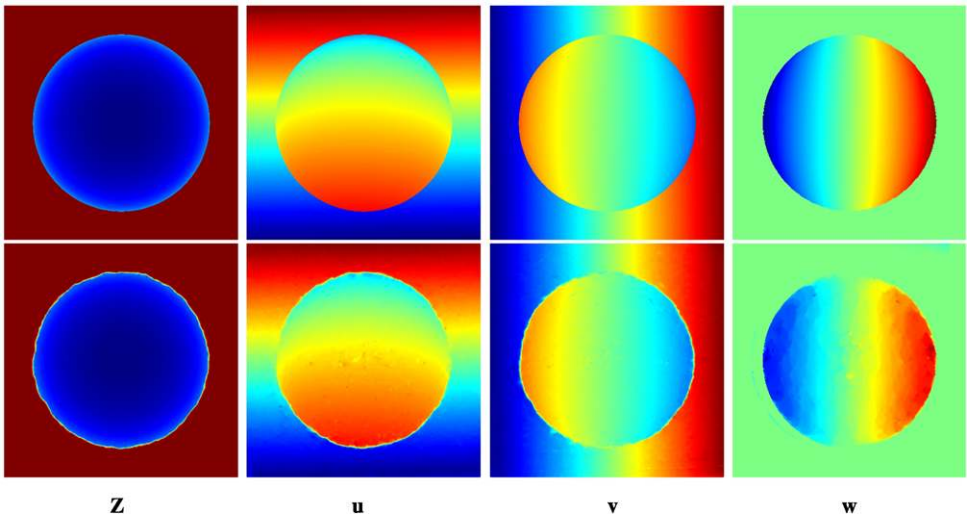

Fig. 7 The top figure represents, from left to right, the ground truth for the depth Z and the 3D motion u, v and w. The bottom figure shows these results computed by our method

Table 2 *Multi-View Rotating Sphere*: The evaluated errors of our computed scene flow and structure over three domains: the continuous regions, the nonoccluded regions, and over all pixels

	$\%\ NRMS_P$	$\%\ NRMS_V$	AAE_V (deg)
w/o Discontinuities	0.65	2.94	1.32
w/o Occlusions	1.99	5.63	2.09
All pixels	4.39	9.71	3.39

our perspective interpretation and correlate with the regions in which the RMS, as we next describe, is relatively high.

For comparison with the results of the scene flow algorithms proposed by Huguet and Devernay (2007), and Wedel et al. (2008, 2011), we project our 3D results, Z, and $\mathbf{V}$, to compute the optical flow and the disparity maps. Despite the differences in the recovered 3D structure and scene flow, which depend on the projection model, the projection of our results should be the same as the given ground truth values for the orthographic projection. However, it is important to note that the resulting errors in our computed optical flow are affected not only by the error in $\mathbf{V}$, but also by the error in Z. Therefore, this comparison is suboptimal for our method.

The results of the disparity and the optical flow are compared with the ground truth values in Fig. 8(b). As can be seen, most of our errors are close to the ball boundary. These errors are probably due to occlusions, large changes in the depth range, and several outliers resulting from error projection.

For quantitative comparisons, we compute the absolute angular error for the optical flow (AAE_{OF}), the *RMS* for the disparity (RMS_d), the optical flow, (RMS_{OF}), and for the optical flow together with the change in the disparity ($RMS_{OF+d'}$). The errors were computed over two domains: the whole sphere and nonoccluded pixels. The computed errors are summarized in Table 3.

The initial disparity map computed by Felzenszwalb and Huttenlocher (2006) was significantly improved by our method (RMS_d decreased from 3.3 to 1.3). This demonstrates the advantages of our method in using the full spatio-temporal information.

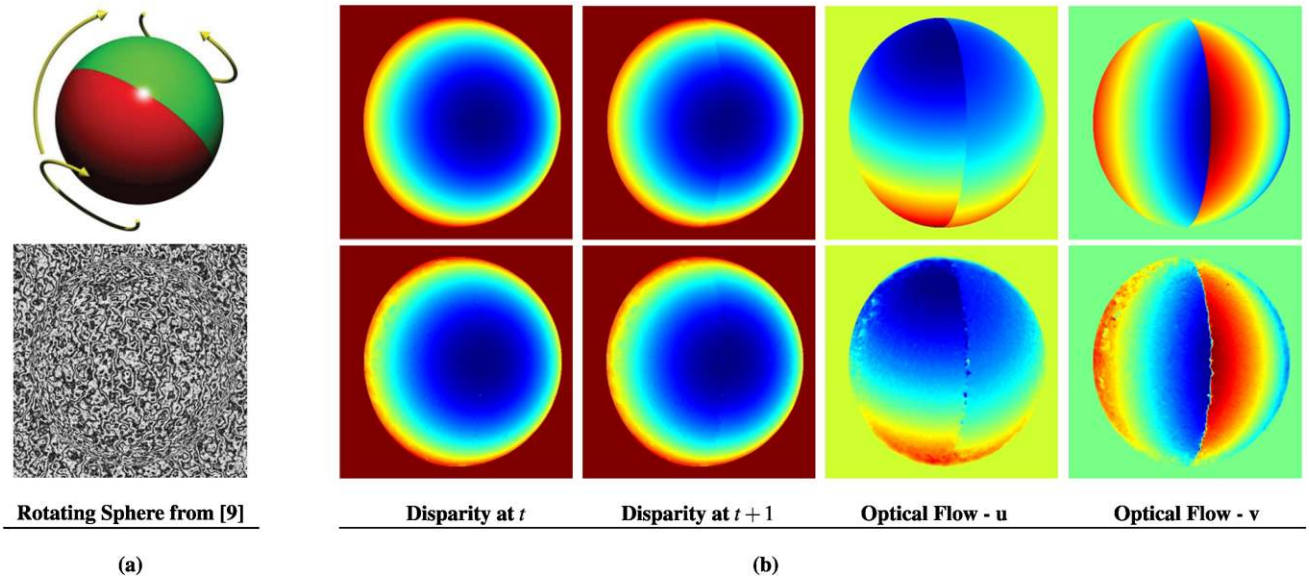

Fig. 8 (**a**) Illustration of the scene motion on *top* and the reference view *below*. (**b**) The *top figure* represents, from *left* to *right*, the ground truth for the disparity at time t, disparity at time $t+1$ and the optical flow, horizontal and vertical components. The *bottom figure* shows these results computed by our method

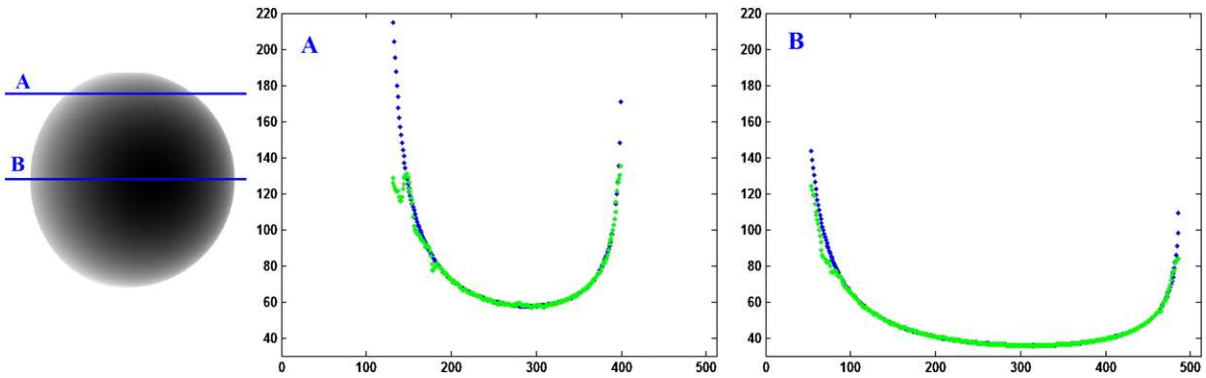

Fig. 9 The *blue plots* are the ground truth shape along longitudinal sections **A** and **B**, marked on the disparity map. The *green plots* is the results of the recovered depth along these sections. Note that due to the perspective projection interpretation of the images, the object shape is not a ball

Table 3 The evaluated errors (w.r.t. the ground truth) of the projection of our scene flow and structure compared with the 2D results of Huguet and Devernay (2007) and Wedel et al. (2008, 2011) as reported in Wedel et al. (2011). RMS (pixels) error in the disparity, optical flow (OF), and optical flow together with the disparity change. Also shown is the absolute angular error (AAE) for the optical flow

Algorithm	RMS_d	Without occluded regions			With occluded regions		
		RMS_{OF}	$RMS_{OF+d'}$	AAE_{OF}	RMS_{OF}	$RMS_{OF+d'}$	AAE_{OF}
Huguet and Devernay (2007)	3.8	0.37	0.83	1.24	0.69	2.51	1.75
Wedel et al. (2011) using SGM	2.9	0.34	0.63	1.04	0.66	2.45	1.50
Our method	1.24	0.32	0.55	1.98	0.43	1.44	2.28

To conclude, our results are similar to results obtained by the state-of-the-art methods on this dataset. Our method is designed to cope with a larger number of views. The results of our method would probably improve if additional views were available.

4.3 Real Data

In this set of experiments we used real-world sequences of a moving scene. These sequences were captured by three USB cameras (IDS uEye UI-1545LE-C). The cameras were

calibrated using the MATLAB Calibration Toolbox. The location of the cameras was fixed for all datasets. All test sequences were taken with an image size of 1280×1024 and then downsampled by half. In all datasets, the depth was initialized to two planes that are parallel to the reference view, located in $Z = 2 \cdot 10^3$ mm and 10^3 mm. We next discuss our results on three datasets.

The first dataset (Fig. 10) involves the rigid 3D motion of a small object (car), in a static scene. The second dataset (Fig. 11) exemplifies a larger motion, mostly in depth direction. The object is low in texture and is moving piecewise rigidly (due to the rotation of the back part of the object). The third experiment consists of a rotating face (Fig. 12). In that case, the 3D motion is generally different for each 3D point. In addition, the motion of the hair is nonrigid. In all three datasets, large occlusions exist due to the notable dissimilarity between the frames.

We present our results in Figs. 10–12. For each dataset we display the magnitude of the estimated scene-flow and the resulting projection of our scene flow onto the reference view. The motion of pixels that are occluded in at least one of the images is indicated by red arrows. Note that most of the errors are found in the computed occluded regions and in the depth discontinuities. In addition, we present the estimated depth masked with the occlusion maps. In order to visually validate our results, we present images warped to the reference view. As can be seen in all the experiments, our method successfully recovers the scene flow and depth. It can be observed that the warped images are very similar to the reference view.

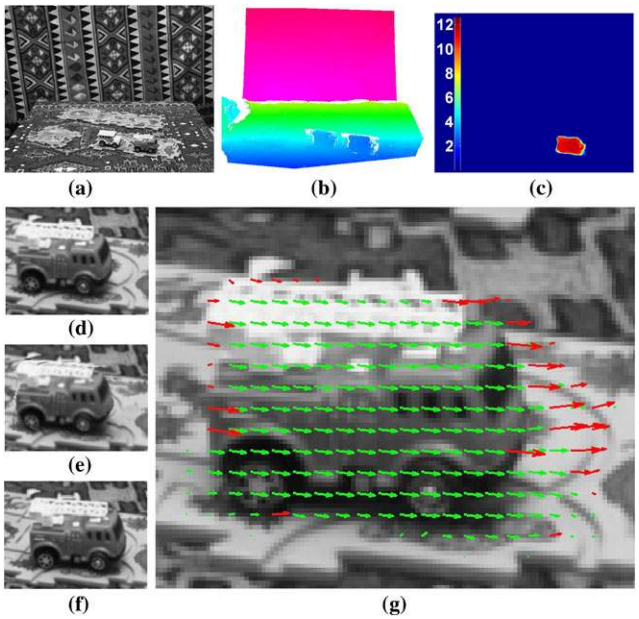

Fig. 10 *Cars dataset*: (**a**) the reference view at time t; (**b**) the depth map masked with the computed occlusion maps; (**c**) the magnitude of the computed scene flow (mm); (**d**) zoom in at time t; (**e**) the corresponding warped image; and (**f**) zoom in at time $t + 1$; (**g**) the projection of the computed scene flow. Occluded pixels are colored in *red*

5 Discussion and Conclusions

In this paper, we proposed a variational approach for simultaneously estimating the scene flow and structure from multi-view sequences. The novel 3D point cloud representation, used to directly model the desired 3D unknowns, allows smoothness assumptions to be imposed directly on the scene flow and structure. In addition, the desired synergy between the 3D unknowns is obtained by imposing the spatio-temporal brightness constancy assumption. Our energy functional explicitly expresses the smoothness and brightness constancy assumptions while enforcing geometric consistency between the views. The redundant informa-

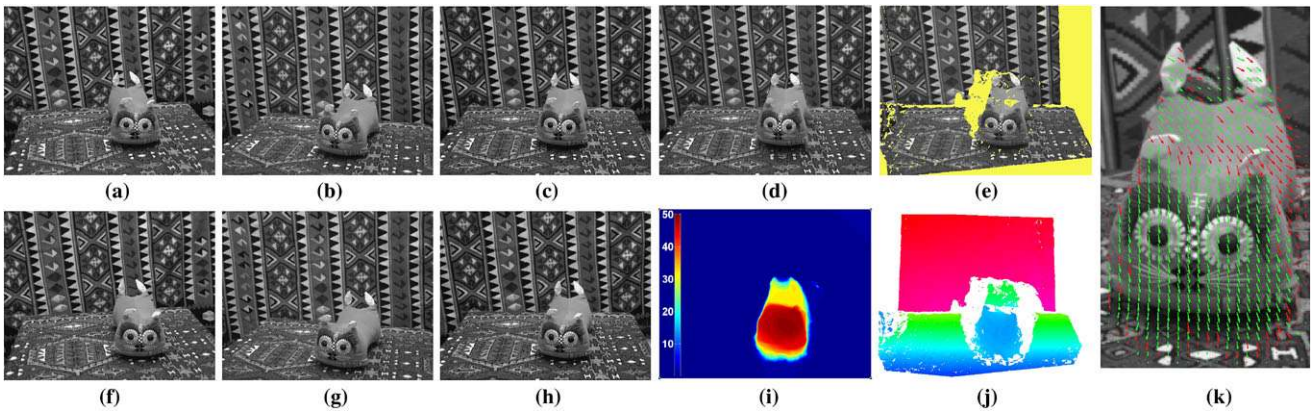

Fig. 11 *Cat dataset*: (**a**–**c**) the three views at time t, where (**c**) is the reference; (**f**–**h**) the corresponding views at time $t + 1$; (**d**) warped image from $\mathbf{h} \rightarrow \mathbf{c}$; (**e**) warped image from $\mathbf{g} \rightarrow \mathbf{c}$, where the yellow regions are the computed occlusions; (**i**) the magnitude of the resulting scene flow (mm); (**j**) the depth map masked by the computed occlusion maps; and (**k**) the projection of the computed scene flow. Occluded pixels are colored in *red*

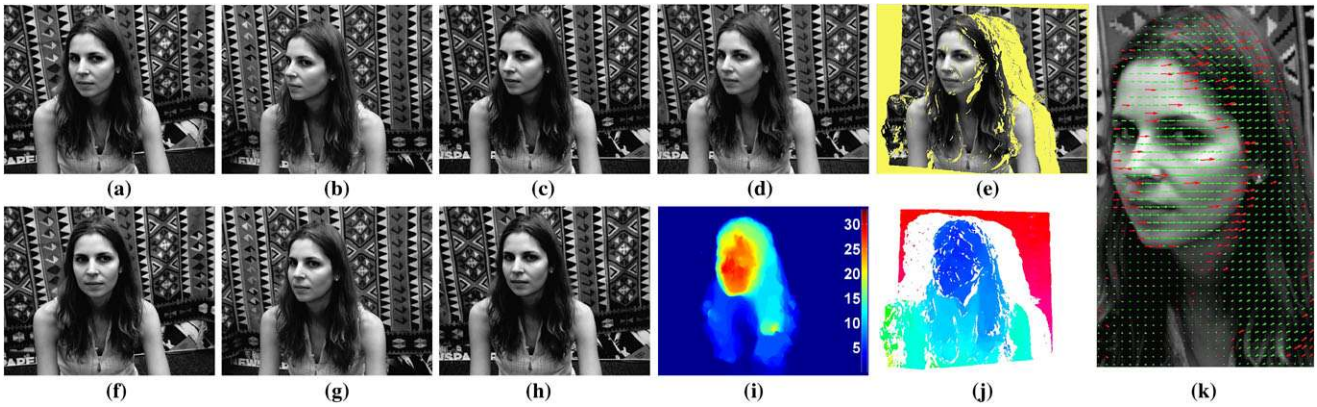

Fig. 12 *Maria dataset*: (**a**–**c**) the three views at time t, where (**c**) is the reference; (**f**–**h**) the corresponding views at time $t+1$; (**d**) warped image from **h** $\rightarrow$ **c**; (**e**) warped image from **f** $\rightarrow$ **c**, where the *yellow regions* are the computed occlusions; (**i**) the magnitude of the resulting scene flow (mm); (**j**) the depth map masked by the computed occlusion maps; and (**k**) the projection of the computed scene flow. Occluded pixels are colored in *red*

tion from multiple views adds supplementary constraints that reduce ambiguities and improve stability.

The combination of our 3D representation in this multi-view variational framework results in a challenging nonconvex optimization problem. Moreover, due to our 3D representation, the relation between the image coordinates and the unknowns is nonlinear (as opposed to optical flow or disparity). Consequently, the derivation of the associated Euler-Lagrange equations involves nontrivial computations. In addition, the use of multiple views requires that occlusions be properly handled since each view adds more occluded regions. Obviously, the occlusion between the views becomes more severe when a wide baseline rig is considered. Our variational framework, which is used for the first time for multiple views and 3D representation, successfully recovers the 3D structure and scene flow despite these difficulties. Our accurate and dense results on real and synthetic data demonstrate the validity of the developed method.

There are several challenges that remain open for future work. These include dealing with larger non-textured regions. Currently, these regions are handled using the regularization, since the data term does not provide sufficient constraints. Another challenge is dealing with occluded regions. Such regions are expected to increase, when the setup consists of even larger differences in the fields of view of the cameras than those considered in our experiments. On the other hand, using more views may provide partial information about these regions. As demonstrated in our results, most of the errors are found in the depth discontinuities and in the occluded regions.

It is, therefore, worthwhile to further study a method that will directly cope with such regions, by, for example, improving the smoothness terms near occlusion boundaries.

Acknowledgements The authors are grateful to the A.M.N. foundation for its generous financial support.

Appendix A: Mapping Between Images

Our 3D parameterization in the presented framework introduces a nonlinear transformation of the 3D unknowns, Z and $\mathbf{V}$, to each of the image's plane. A notable challenge in the minimization of the proposed functional arises from the nontrivial mapping of the images' coordinates to the reference camera coordinate system.

Using our parametrization, each pixel in the reference camera, (x, y), and its corresponding depth, $Z(x, y)$, specify a 3D point, $\mathbf{P}$ (see Eq. (5)). It follows that projecting $\mathbf{P}$ onto the ith camera maps $(x, y, Z(x, y))$ to the point $\mathbf{p}_i = (x_i, y_i)^T$. That is,

$$\mathbf{p}_i = Proj\big(\mathbf{P}, M^i\big) = f^i\big(x, y, Z(x, y)\big), \tag{17}$$

where f^i is the mapping to the corresponding ith image. More precisely, f^i is given by substituting Eq. (5) into Eq. (1). For example, the component x_i is given by:

$$x_i = \frac{a \cdot Z + b}{c \cdot Z + d}\ . \tag{18}$$

The coefficients a, b, c and d depend on the reference camera coordinates, (x, y):

$$\begin{aligned} a(x, y) &= M^i_{11} \cdot (x/s_x - o_x/s_x) + M^i_{12} \cdot (y/s_y - o_y/s_y) \\ &\quad + M^i_{13}, \\ b(x, y) &= M^i_{14}, \\ c(x, y) &= M^i_{31} \cdot (x/s_x - o_x/s_x) + M^i_{32} \cdot (y/s_y - o_y/s_y) \\ &\quad + M^i_{33}, \\ d(x, y) &= M^i_{34}, \end{aligned} \tag{19}$$

where M^i is the 3×4 projection matrix of the ith camera (subscripts denote the row and column indices). The expression for y_i is equivalently computed.

Similarly, at time step $t+1$, projecting $\widehat{\mathbf{P}} = \mathbf{P} + \mathbf{V}$ maps $(x, y, Z(x, y), V(x, y))$ to $\widehat{\mathbf{p}}_i$, denoted by a mapping, $\widehat{f}^i$:

$$\widehat{\mathbf{p}}_i = Proj\big(\widehat{\mathbf{P}}, M^i\big) = \widehat{f}^i\big(x, y, Z(x, y), \mathbf{V}(x, y)\big). \tag{20}$$

Analogously to Eq. (18), the component $\widehat{x}_i$ is given by:

$$\widehat{x}_i = \frac{a \cdot Z + M^i_{11} \cdot u + M^i_{12} \cdot v + M^i_{13} \cdot w + b}{c \cdot Z + M^i_{31} \cdot u + M^i_{32} \cdot v + M^i_{33} \cdot w + d}, \tag{21}$$

where the coefficients a, b, c and d are defined in Eq. (19).

Appendix B: Image Derivatives with Respect to the 3D Unknowns

A first step toward the numerical solution of the resulting Euler-Lagrange equations (Eq. (12) or Eq. (13)) requires computing the derivatives of the intensity functions with respect to the 3D unknowns. To produce the final expressions for these derivatives, the nonlinear relation between the 3D unknowns and the image plane has to be carefully considered (see Appendix A). This appendix shows how these computations are performed. The mathematical analysis is preformed in the continuous domain. Thus, the frames as well as the 3D unknowns are regarded as continuous functions. Finally, the resulting equations are discretized by using standard approximations for the derivatives.

For simplicity, given a time step, t, we use the intensity functions I_i^t and I_i^{t+1} to abbreviate $I_i(\mathbf{p}_i, t)$ and $I_i(\widehat{\mathbf{p}}_i, t+1)$, respectively. We next elaborate on the computation of derivatives of I_i^t and I_i^{t+1} with respect to Z and u, denoted by $\partial_Z I_i^t$, $\partial_Z I_i^{t+1}$ and $\partial_u I_i^{t+1}$ (the other derivatives with respect to v and w are similarly computed).

I_i^t can be regarded as a function of the reference image coordinates, (x, y), and the corresponding depth, $Z(x, y)$, by considering a composition of two functions: the ith intensity function and the mapping transformation, defined in Appendix A. That is,

$$I_i^t\big(x, y, Z(x, y)\big) = I_i(f^i\big(x, y, Z(x, y), t\big). \tag{22}$$

Similarly, I_i^{t+1} can be regarded as a function of $(x, y, Z(x, y))$ and $\mathbf{V}$. That is,

$$I_i^{t+1}(x, y, Z, \mathbf{V}) = I_i\big(\widehat{f}^i(x, y, Z, \mathbf{V}), t+1\big). \tag{23}$$

Considering Eqs. (17)–(20), the chain rule is applied for computing the partial derivatives:

$$\partial_Z I_i^t = \big(\nabla I_i^t\big)^T \cdot \partial_Z \mathbf{p}_i, \tag{24}$$

$$\partial_Z I_i^{t+1} = \big(\nabla I_i^{t+1}\big)^T \cdot \partial_Z \widehat{\mathbf{p}}_i, \tag{25}$$

$$\partial_u I_i^{t+1} = \big(\nabla I_i^{t+1}\big)^T \cdot \partial_u \widehat{\mathbf{p}}_i. \tag{26}$$

The derivatives $\partial_Z \mathbf{p}_i^T = (\partial_Z x_i, \partial_Z y_i)^T$ are directly computed from Eqs. (18)–(19).

To compute the derivative of I_i^t with respect to $\mathbf{p}_i$, $(\nabla I_i^t)^T$, we use a warping approach. As discussed in Appendix A, a nonlinear mapping relates each of the image's plane to the reference camera. By warping I_i^t toward the reference image using the estimated Z, the values of I_i^t can be directly related to the reference image values, I_0^t. Specifically, the required derivatives, ∇I_i^t are then computed using the warped image. Let $I_{i,w}^t$ be the warped image of I_i^t. That is,

$$I_{i,w}^t(x, y) = I_i(\mathbf{p}_i, t). \tag{27}$$

The warped image gradient is related to the original image by:

$$\big(\nabla I_{i,w}^t\big)^T = \big(\partial_x I_{i,w}^t, \partial_y I_{i,w}^t\big) = \big(\nabla I_i^t\big)^T \cdot \underbrace{\begin{pmatrix} \frac{\partial x_i}{\partial x} & \frac{\partial x_i}{\partial y} \\ \frac{\partial y_i}{\partial x} & \frac{\partial y_i}{\partial y} \end{pmatrix}}_{J}, \tag{28}$$

where J is the Jacobian matrix of the change of coordinates, $(x_i, y_i) \rightarrow (x, y)$. Therefore, the original image derivatives are obtained by multiplying Eq. (28) by J^{-1}, leading to:

$$\big(\nabla I_i^t\big)^T = \big(\nabla I_{i,w}^t\big)^T \cdot J^{-1}. \tag{29}$$

The Jacobian matrix, J, is obtained by computing the derivatives of $\mathbf{p}_i$ with respect to x and y. In particular, J involves the derivatives of $Z(x, y)$, namely $\partial_x Z$ and $\partial_y Z$. Following the explanation above, $\nabla I_{i,w}^{t+1}$ is similarly computed. In this case, the Jacobian matrix, J, additionally involves the derivatives of u, v and w with respect to the reference camera coordinates.

Appendix C: Linearizion

This appendix describes the linearizion process of the resulting Euler-Lagrange equations and the numerical approximations used. At each pyramid level, a linear system of equations is obtained and small increments in the 3D unknowns, dZ, and $d\mathbf{V}$, are estimated. The total solution, $Z + dZ$, and $\mathbf{V} + d\mathbf{V}$, is then used to initialize the next finer level (see Sect. 2.3.2).

Considering equations (12)–(13), there are two sources of nonlinearity:

1. nonlinearized data term;
2. nonquadratic cost function Ψ.

Following the numerical approach suggested by Brox et al. (2004), two nested fixed point iterations are used at each pyramid level to remove the nonlinearity.

The outer iteration is responsible for removing the nonlinearity resulting from the nonlinear data term, using fixed point iteration on Z and $\mathbf{V}$. Let k be the outer index iteration. The solution at the $(k+1)$th iteration is decomposed of the previous solution and small, unknown increments. That is, $Z^{k+1} = Z^k + dZ^k$ and $\mathbf{V}^{k+1} = \mathbf{V}^k + d\mathbf{V}^k$, where $d\mathbf{V}^k = (du^k, dv^k, dw^k)^T$.

The first step toward linearizion is approximating the nonlinear expression given in Eq. (11) using first order Taylor expansion. We use Δ_i^k, $\widehat{\Delta}_i^k$ and $\Delta_i^{t,k}$ to denote the expressions given in Eq. (11) using the fixed values Z^k and $\mathbf{V}^k$. That is,

$$\begin{aligned}
\Delta_i^k &= I_i\big(\mathbf{p}_i^k, t\big) - I_0\big(\mathbf{p}_0^k, t\big),\\
\widehat{\Delta}_i^k &= I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) - I_0\big(\widehat{\mathbf{p}}_0^k, t+1\big),\\
\Delta_i^{t,k} &= I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) - I_i\big(\mathbf{p}_i^k, t\big),
\end{aligned} \tag{30}$$

where $\mathbf{p}_i^k = Proj(\mathbf{P}^k, M^i)$ and $\mathbf{P}^k$ is given by placing Z^k in Eq. (5). The expressions for $\widehat{\mathbf{p}}_i^k$ and $\widehat{\mathbf{P}}^k$ are analogously given. Using these notations, the first order Taylor expansions for these expressions are given by:

$$\begin{aligned}
\Delta_i^{k+1} &\approx \Delta_i^k + \partial_Z \Delta_i^k \cdot dZ^k,\\
\widehat{\Delta}_i^{k+1} &\approx \widehat{\Delta}_i^k + \partial_Z \widehat{\Delta}_i^k \cdot dZ^k\\
&\quad + \partial_u \widehat{\Delta}_i^k \cdot du^k + \partial_v \widehat{\Delta}_i^k \cdot dv^k + \partial_w \widehat{\Delta}_i^k \cdot dw^k,\\
\Delta_i^{t,k+1} &\approx \Delta_i^t + \partial_Z \Delta_i^{t,k}\\
&\quad + \partial_u \Delta_i^{t,k} \cdot du^k + \partial_v \Delta_i^{t,k} \cdot dv^k + \partial_w \Delta_i^{t,k} \cdot dw^k.
\end{aligned} \tag{31}$$

Equation (31) is computed by using the first order Taylor expansion for the following expressions:

$$\begin{aligned}
I_i\big(\mathbf{p}_i^{k+1}, t\big) &= I_i\big(Proj\big(\mathbf{P}^{k+1}, M^i\big), t\big)\\
&\approx I_i\big(\mathbf{p}_i^k, t\big) + \partial_Z I_i(\mathbf{p}_i, t) \cdot dZ^k,
\end{aligned} \tag{32}$$

$$\begin{aligned}
I_i\big(\widehat{\mathbf{p}}_i^{k+1}, t+1\big) &= I_i\big(Proj\big(\widehat{\mathbf{P}}^{k+1}, M^i\big), t+1\big)\\
&\approx I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) + \partial_Z I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) \cdot dZ^k\\
&\quad + \partial_u I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) \cdot du^k\\
&\quad + \partial_v I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) \cdot dv^k\\
&\quad + \partial_w I_i\big(\widehat{\mathbf{p}}_i^k, t+1\big) \cdot dw^k,
\end{aligned} \tag{33}$$

where $\mathbf{P}^{k+1} = \mathbf{P}^k + d\mathbf{P}^k$ is given by placing $Z^k + dZ^k$ (Eq. (5)). Similarly, $\widehat{\mathbf{P}}^{k+1} = \widehat{\mathbf{P}}^k + d\widehat{\mathbf{P}}^k$ where $d\widehat{\mathbf{P}}^k = dZ^k + d\mathbf{V}^k$. The computation of the image derivatives with respect to the 3D unknowns is detailed in Appendix A.

Therefore, deriving the associated Euler-Lagrange equations with respect to the unknown increments dZ^k and du^k results in:

$$\begin{aligned}
0 &= \sum_{i=0}^{N-1} \Psi'\big(\big(\Delta_i^{t,k+1}\big)^2\big)\Delta_i^{t,k+1} \cdot \big(\Delta_i^{t,k}\big)_Z\\
&\quad + \sum_{i=1}^{N-1} \Psi'\big(\big(\Delta_i^{k+1}\big)^2\big)\Delta_i^{k+1} \cdot \big(\Delta_i^k\big)_Z\\
&\quad + \sum_{i=1}^{N-1} \Psi'\big(\big(\widehat{\Delta}_i^{k+1}\big)^2\big)\widehat{\Delta}_i^{k+1} \cdot \big(\widehat{\Delta}_i^k\big)_Z\\
&\quad - \alpha\mu \cdot \mathrm{div}\big(\Psi'\big(\big|\nabla Z^{k+1}\big|^2\big)\nabla Z^{k+1}\big),
\end{aligned} \tag{34}$$

$$\begin{aligned}
0 &= \sum_{i=0}^{N-1} \Psi'\big(\big(\Delta_i^{t,k+1}\big)^2\big)\Delta_i^{t,k+1} \cdot \big(\Delta_i^{t,k}\big)_u\\
&\quad + \sum_{i=1}^{N-1} \Psi'\big(\big(\widehat{\Delta}_{i^{k+1}}\big)^2\big)\widehat{\Delta}_i^{k+1} \cdot \big(\widehat{\Delta}_i^k\big)_u\\
&\quad - \alpha \cdot \mathrm{div}\big(\Psi'\big(\big|\nabla u^{k+1}\big|^2 + \big|\nabla v^{k+1}\big|^2\\
&\quad + \big|\nabla w^{k+1}\big|^2\big)\nabla u^{k+1}\big).
\end{aligned} \tag{35}$$

The dependency of the above two equations in the increments, dZ^k and du^k, is obtained by substituting Eq. (31) into Δ_i^{k+1}, $\Delta_i^{t,k+1}$, and, $\widehat{\Delta}_i^{k+1}$. The equations for dv^k and dw^k are similar to Eq. (35).

Applying the above approximations (Eq. (31)), the resulting Euler-Lagrange equations are a nonlinear system of equations in the unknowns dZ^k and $d\mathbf{V}^k$. The remaining nonlinearity is originated by Ψ'. Therefore, an additional fixed point iterations loop for Ψ' expressions is preformed. Finally, after standard discretization of the derivatives, a linear system of equations is introduced. The solution is obtained by applying the *successive overrelaxation* (SOR) method.

References

Ayvaci, A., Raptis, M., & Soatto, S. (2010). Occlusion detection and motion estimation with convex optimization. *NIPS* (pp. 100–108).

Basha, T., Moses, Y., & Kiryati, N. (2010). Multi-view scene flow estimation: A view centered variational approach. In *Proc. IEEE conf. comp. vision patt. recog.* (pp. 1506–1513).

Ben-Ari, R., & Sochen, N. A. (2007). Variational stereo vision with sharp discontinuities and occlusion handling. In *Proc. int. conf. comp. vision* (pp. 1–7).

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Proc. European conf. comp. vision* (pp. 25–36).

Carceroni, R. L., & Kutulakos, K. N. (2002). Multi-view scene capture by surfel sampling: from video streams to non-rigid 3d motion, shape and reflectance. *International Journal of Computer Vision*, *49*(2–3), 175–214.

Courchay, J., Pons, J. P., Monasse, P., & Keriven, R. (2009). Dense and accurate spatio-temporal multi-view stereovision. In *Asian conf. on computer vision* (pp. 11–22).

Felzenszwalb, P., & Huttenlocher, D. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, *70*(1), 41–54.

Furukawa, Y., & Ponce, J. (2008). Dense 3d motion capture from synchronized video streams. In *Proc. IEEE conf. comp. vision patt. recog.*

Huguet, F., & Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. In *Proc. int. conf. comp. vision* (pp. 1–7).

Isard, M., & MacCormick, J. (2006). Dense motion and disparity estimation via loopy belief propagation. In *Asian conf. on computer vision* (Vol. 3852, p. 32).

Li, R., & Sclaroff, S. (2008). Multi-scale 3d scene flow from binocular stereo sequences. *Computer Vision and Image Understanding*, *110*(1), 75–90.

Min, D. B., & Sohn, K. (2006). Edge-preserving simultaneous joint motion-disparity estimation. In *Proc. international conf. patt. recog.* (pp. 74–77).

Neumann, J., & Aloimonos, Y. (2002). Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, *47*(1–3), 181–193.

Pock, T., Schoenemann, T., Graber, G., Bischof, H., & Cremers, D. (2008). A convex formulation of continuous multi-label problems. In *Proc. European conf. comp. vision* (pp. 792–805).

Pons, J., Keriven, R., & Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, *72*(2), 179–193.

Robert, L., & Deriche, R. (1996). Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In *Proc. European conf. comp. vision* (pp. 439–451).

Scharstein, D., & Szeliski, R. Middlebury stereo vision research page. http://vision.middlebury.edu/stereo.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1–3), 7–42.

Scharstein, D., & Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Proc. IEEE conf. comp. vision patt. recog.* (pp. 195–202).

Strecha, C., Tuytelaars, T., & Gool, L. J. V. (2003). Dense matching of multiple wide-baseline views. In *Proc. int. conf. comp. vision* (pp. 1194–1201).

Vedula, S., Baker, S., Rander, P., Collins, R. T., & Kanade, T. (1999). Three-dimensional scene flow. In *Proc. int. conf. comp. vision* (pp. 722–729).

Vedula, S., Baker, S., Rander, P., Collins, R. T., & Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 475–480.

Vedula, S., Baker, S., Seitz, S., & Kanade, T. (2000). Shape and motion carving in 6D. In *Proc. IEEE conf. comp. vision patt. recog.* (Vol. 2).

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., & Cremers, D. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, *95*(1), 29–51.

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., & Cremers, D. (2008). Efficient dense scene flow from sparse or dense stereo data. In *Proc. European conf. comp. vision* (pp. 739–751).

Woodford, O. J., Torr, P. H. S., Reid, I. D., & Fitzgibbon, A. W. (2009). Global stereo reconstruction under second-order smoothness priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(12), 2115–2128.

Young, D. (1954). Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, *76*(1), 92–111.

Zhang, Y., & Kambhamettu, C. (2000). Integrated 3d scene flow and structure recovery from multiview image sequences. In *Proc. IEEE conf. comp. vision patt. recog.* (Vol. 2, pp. 674–681).

Zhang, Y., & Kambhamettu, C. (2001). On 3d scene flow and structure estimation. In *Proc. IEEE conf. comp. vision patt. recog.* (pp. 778–785).