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

Tali Basha
Tel Aviv University
Tel Aviv 69978, Israel
talib@eng.tau.ac.il

Yael Moses
The Interdisciplinary Center
Herzliya 46150, Israel
yael@idc.ac.il

Nahum Kiryati
Tel Aviv University
Tel Aviv 69978, Israel
nk@eng.tau.ac.il

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. The minimization of the functional is successfully obtained despite the non-convex optimization problem. The proposed method was tested on real and synthetic data.

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, motion analysis, tracking, navigation, 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 non-rigid 3D scene (Vedula $et\ al.$ [17]). It follows directly from this definition that 3D recovery of the surface must be an essential part of scene flow algorithms, 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 variational method used.

Most existing methods for scene flow and surface estimation parameterize the problem in 2D rather than 3D. That is, they compute the projection of the desired 3D unknowns, namely disparity and optical flow (e.g., [22, 23, 18, 8, 10, 7, 20, 9]). Using 3D parametrization allows us to impose primary assumptions on the unknowns prior to their projection. For example, a constant 3D motion field of a scene may project to a discontinuous 2D field. Hence, in this example, smoothness assumptions hold for 3D parametrization but not for 2D one. We propose a 3D point cloud parametrization of the 3D structure and 3D motion. That is, for each pixel in a reference view, a depth value and a 3D motion vector are computed. Our 3D parametrization allows direct extension to multiple views, without changing the problem's dimension.

Decoupling the spatio-temporal information leads to sequential estimation of scene flow and structure (e.g., [17, 18, 22, 23, 3, 12, 20]). Such methods rely on pre-computed motion or structure results and do not utilize the full spatio-temporal information. For example, Vedula $et\ al.$ [18] suggested independent computation of the optical flow field for each camera without imposing consistency between the flow fields. Wedel $et\ al.$[20] enforced consistency on the stereo and motion solutions. However, the disparity map is separately computed, and thus the results are still sensitive to its errors. To overcome these limitations, simultaneous recovery of the scene flow and structure was suggested (e.g., [19, 8, 10, 7, 11]). However, most of these methods suffer from the restriction of using 2D parametrization; in particular, they are limited to two views (3D ones are discussed in Sec. 1.1). Our method involves multi-view information that improves stability and reduces ambiguities.

We suggest coupling the spatio-temporal information from a set of sequences using 3D parameterization for solving the problem. To do so, a global energy functional is defined to incorporate the multi-view geometry with a bright-

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

ness 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 non-quadratic cost functions. This approach is motivated by the state-of-the-art optical flow variational approach of Brox *et al.* [2]. Our method is the first to extend it to multiple views and 3D parametrization. The minimization of the resulting non-convex 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 suggest a new evaluation based on the 3D errors. We argue that the conventional 2D error used for evaluating stereo and optical flow algorithms does not necessarily correlate with the suggested 3D error. In particular, we claim that the ranking of stereo algorithms (e.g., [15]) may vary when the 3D errors are considered.

The main contribution of this paper is the combination of a novel 3D formulation and 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. We manage to successfully minimize the functional despite the challenging non-convex optimization problem.

The rest of the paper is organized as follows. We begin with reviewing related studies in Sec. 1.1. Sec. 2 describes our method. Sec. 3 provides an insight to our quantitative 3D evaluation measures. In Sec. 4 we present the experimental results. We discuss our conclusions in Sec. 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 [17], various mesh representations [6, 4, 11] and dynamic surfels [3]. In contrast to our method, each of these 3D representations can provide a complete, view-independent 3D description of the scene. However, using these methods, the type of 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, 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 [7, 20]. Huguet & Devernay [7] proposed to simultaneously compute the optical flow field and two disparity maps (in successive time steps), while Wedel *et al.* [20] decoupled the disparity at the first time step from the rest of the computation. Both extend the variational framework of Brox *et al.* [2] 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. In addition, their methods were not extended to multiple views.

A multi-view energy minimization framework was presented by Zhang & Kambhamettu [22]. 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. In their method, 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.* [12]. 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.

There are only few multi-view methods that use 3D representations and simultaneously solve the 3D surface and motion. Neumann & Aloimonos [11] 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 & Ponce [6] constructed an initial polyhedral mesh at the first frame. It is tracked assuming locally rigid motion and successively, globally non-rigid deformation. Courchay *et al.* [4] 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.* [4] and Furukawa & Ponce [6] 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 sequence 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, 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:

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

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 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. 1).

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

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

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 (similar parametrization for stereo was used by Robert & Deriche.[13]). That is,

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

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

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 $\widehat{\mathbf{P}}(x, y, t)$, which is projected to a different image pixel $\widehat{\mathbf{p}} = (\widehat{x}, \widehat{y})^T$ (unless there is no motion).

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

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., [7, 20, 9], where additional cameras require additional sets of unknowns (e.g., optical flow or disparity field). Moreover, our representation does not require image rectification.

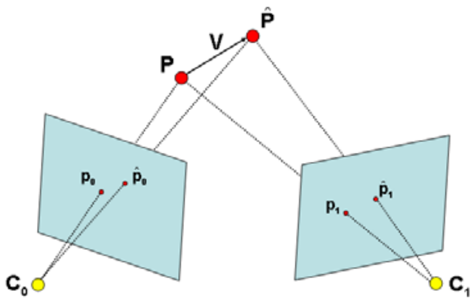

Figure 1. The point $\mathbf{P}$ is projected to pixels $\mathbf{p}_0$ and $\mathbf{p}_1$ on cameras C_0 and C_1, respectively. The new 3D location at $t+1$ is given by $\widehat{\mathbf{P}} = \mathbf{P} + \mathbf{V}$ and it is projected to $\widehat{\mathbf{p}}_0$ and $\widehat{\mathbf{p}}_1$.

2.2. The Energy Functional

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

$$E(Z, \mathbf{V}) = E_{data} + \alpha E_{smooth}. \qquad (5)$$

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 Sec. 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 assumptions: The data term imposes the BC 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 BC 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 BC assumption before and after 3D displacement; BC_{s_1} and BC_{s_2} penalize deviation from the BC assumption between the reference image and each of the other views at time t and $t+1$, respectively. Formally the penalizers for each pixel are defined by:

$$\begin{aligned} BC_m(Z, \mathbf{V}) &= \sum_{i=0}^{N-1} c_m^i \Psi(|I_i(\mathbf{p}_i, t) - I_i(\widehat{\mathbf{p}}_i, t+1)|^2), \\ BC_{s_1}(Z) &= \sum_{i=1}^{N-1} c_{s_1}^i \Psi(|I_0(\mathbf{p}_0, t) - I_i(\mathbf{p}_i, t)|^2), \qquad (6) \\ BC_{s_2}(Z, \mathbf{V}) &= \sum_{i=1}^{N-1} c_{s_2}^i \Psi(|I_0(\widehat{\mathbf{p}}_0, t+1) - I_i(\widehat{\mathbf{p}}_i, t+1)|^2), \end{aligned}$$

where c_*^i is a binary function that omits occluded pixels from the computation (see Sec. 2.4) and $\Psi(s^2)$ is a chosen cost function. We use a non-quadratic robust cost function $\Psi(s^2) = \sqrt{s^2+\epsilon^2}, (\epsilon = 0.0001)$, which is a smooth approximation of L_1 (see [2]), 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, or occlusions. In this formulation, no linear approximations are made; hence large displacements between frames are allowed. Note that we chose not to impose an additional gradient constancy assumption. Previous studies for estimating optical flow (e.g., [2]) or scene flow (e.g.,[7]) imposed this assumption for improved robustness against illumination changes. Nevertheless, since the gradient is viewpoint dependent, this assumption does not hold in the spatial domain.

Smoothness assumptions: Piecewise smoothness assumptions are imposed on both the 3D motion field and surface. 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:

$$\begin{aligned} S_m(\mathbf{V}) &= \Psi(|\nabla u(x,y,t)|^2 + |\nabla v(x,y,t)|^2 + |\nabla w(x,y,t)|^2), \\ S_s(Z) &= \Psi(|\nabla Z(x,y,t)|^2), \end{aligned} \tag{7}$$

where S_m is the penalizer of deviation from the motion smoothness assumption and S_s is the penalizer for shape. 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 [21] might be more suitable in our framework.

The total energy function is obtained by integrating the penalty (Eq. 6-7) over all pixels in the reference camera, Ω:

$$E(Z, \mathbf{V}) = \int_{\Omega} [\underbrace{BC_m + BC_s}_{data} + \alpha \underbrace{(S_m + \mu S_s)}_{smooth}] dxdy, \tag{8}$$

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

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:

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

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

$$\begin{aligned} &\sum_{i=0}^{N-1} \Psi'((\Delta_i^t)^2)\Delta_i^t \cdot (\Delta_i^t)_Z + \sum_{i=1}^{N} \Psi'((\Delta_i)^2)\Delta_i \cdot (\Delta_i)_Z + \\ &\sum_{i=1}^{N-1} \Psi'((\widehat{\Delta}_i)^2)\widehat{\Delta}_i \cdot (\widehat{\Delta}_i)_Z \; - \alpha\mu \cdot div(\Psi'(|\nabla Z|^2)\nabla Z) = 0, \end{aligned} \tag{9}$$

$$\begin{aligned} &\sum_{i=0}^{N-1} \Psi'((\Delta_i^t)^2)\Delta_i^t \cdot (\Delta_i^t)_u + \sum_{i=1}^{N} \Psi'((\widehat{\Delta}_i)^2)\widehat{\Delta}_i \cdot (\widehat{\Delta}_i)_u \\ &- \alpha \cdot div(\Psi'(|\nabla u|^2 + |\nabla v|^2 + |\nabla w|^2)\nabla u) = 0. \end{aligned} \tag{10}$$

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. 10 due to the symmetry of these variables.

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.

Due to space limitations, the detailed expressions for the Euler-Lagrange equations are not represented. However, it is clear from Eq. 9 or Eq. 10 that the images are non-linear 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 non-trivial manner. We refer the reader to our technical report [1] for the detailed description.

2.3.2 Numerics

Our parametrization and functional represent precisely the desired model (no approximations are made), resulting in a challenging minimization problem. In particular, the use of non-linearized data terms and non-quadratic penalizers yields a non-linear system in the four unknowns Z and $\mathbf{V}$ (e.g., Eq. 9-10). Moreover, one has to deal with the problem of multiple local minima as a result of the non-convex functional. In our method, the derivation and discretization of the equations results in additional complexity since the perspective projection is non-linear in the unknowns Z and $\mathbf{V}$.

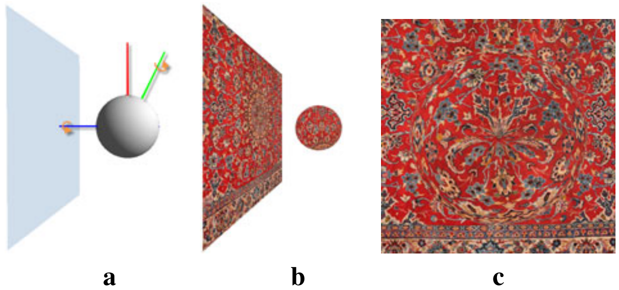

Figure 2. **(a)** Illustration of the rotation axes. The sphere is rotating around the green axis and the plane around the red one. **(b)** With texture. **(c)** The reference view before rotation.

We cope with these difficulties by using a multi-resolution warping method coupled with two nested fixed point iterations as previously suggested by [2]. 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. 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 at each level 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 $\alpha_\ell = \alpha \cdot \eta^\ell$ w.r.t the pyramid level, ℓ.

The solution in a given pyramid level is obtained from two nested fixed point iterations that are responsible to remove the nonlinearity in the equations. The outer iteration accounts for the linearizion of the data term. Using the first order Taylor expansion, at each outer iteration, k, small increments in the solutions, dZ^k and $d\mathbf{V}^k$ 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 re-warped accordingly and the images derivatives are re-computed. 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. The final linear system is solved by applying the *successive overrelaxation* (SOR) method. We refer to [1] for additional details on the numeric approach presented in this section.

2.4. Occlusions

Occlusions are computed by determining the visibility of each 3D surface point in each of the cameras at each time step. Clearly, 3D points that are occluded in a specific image do not satisfy the BC assumption. Hence, the associated component of the data term should be omitted. This is accomplished by computing for each view (other than the reference) three occlusion maps (c_*^i). Each of the three maps corresponds to the relevant penalizer in the data term (Eq. 6). The computed maps are used as 2D binary functions, multiplying respectively each of the data term components. A modified Z-buffering is used for estimating the occlusion maps. 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

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 (the disparity or the optical flow). We suggest a new evaluation by assessing the direct error in the recovered 3D surface and the 3D motion map. That is, we compute the deviation of the estimated 3D point, $\mathbf{P}(x, y)$, from its ground truth, $\mathbf{P}_o(x, y)$. Various statistics over these errors can be chosen. We compute the *normalized root mean square* (*NRMS*) error, which is the percentage of the *RMS* error from the range of the observed values. We define $NRMS_\mathbf{P}$ by:

$$NRMS_\mathbf{P} \quad = \quad \frac{\sqrt{\frac{1}{N}\sum_\Omega ||\mathbf{P}(x,y)^T - \mathbf{P}_o(x,y)^T||^2}}{max(||\mathbf{P}_o(x,y)||) - min(||\mathbf{P}_o(x,y)||)}, \qquad (11)$$

where Ω denotes the integration domain (e.g., non-occluded areas) and N is the number of pixels. Similarly, $NRMS_\mathbf{V}$ 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 $\mathbf{V}$.

The proposed evaluation is motivated by the observation that the errors in 2D (in the image plane) do not necessarily correlate with the errors in 3D. That is, the 2D error at a given pixel depends not only on the magnitude of the 3D error but also on the position of the 3D point relative to the camera and on the 3D error direction . Thus, when comparing the results of 3D reconstruction or scene flow algorithms, using the 3D evaluation may result in different ranking than when using 2D errors. To test this observation, we compared the results of various statistics computed using 2D and 3D errors on the top five ranked stereo algorithms in Middlebury datasets [14]. The results demonstrate that changes in the ranking indeed occur when RMS is considered.

4. Experimental Results

To assess the quality and accuracy of our method, we preformed experiments on synthetic and real data. Our algorithm was implemented in C using the *OpenCV* library.

Like all variational methods, our method requires initial depth and 3D motion maps. In all experiments the 3D motion field was simply initiated to zero. In the first two experiments we used the stereo algorithm proposed in [5] to obtain an initial depth map between the reference camera and one of the other views. In the third experiment, we used a naive initialization of two parallel planes. This initialization is very far from the real depth. We next elaborate on each of the experiments.

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 egomotion of the cameras. The Middlebury stereo datasets, $Cones$, $Teddy$ and $Venus$ [16], were used for generating the data (as in [7]). Each of the datasets consists of 9 rectified images taken from equally spaced viewpoints. 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.

For comparison with the results of the scene flow algorithm proposed by Huguet $et\ al.$[7], 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 normalized root mean square error (*NRMS*) for the optical flow and each of the disparity fields at time t and $t+1$. These measurements are given in Table. 1.

We achieved significantly better results for the optical flow and disparity at time $t+1$. There is an improvement of 46%-54% in the *NRMS* error of the optical flow and 28%-58% in the *NRMS* 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.

		NRMS (%)			*AAE*
		O.F.	disp. at t	disp. at $t+1$	*(deg)*
Cones	4 Views	1.32	6.22	6.23	0.12
	2 Views	3.07	6.52	6.55	0.39
	[7]	5.79	5.55	13.79	0.69
Teddy	4 Views	2.53	6.13	6.15	0.22
	2 Views	2.85	7.04	7.11	1.01
	[7]	6.21	5.64	17.22	0.51
Venus	4 Views	1.55	5.39	5.39	1.09
	2 Views	1.98	6.36	6.36	1.58
	[7]	3.70	5.79	8.84	0.98

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 $et\ al.$[7]. Normalized RMS (NRMS) error in the optical flow (O.F.), disparity at time t, and the disparity at time $t+1$. Also shown, the absolute angular error (AAE) corresponding to the optical flow.

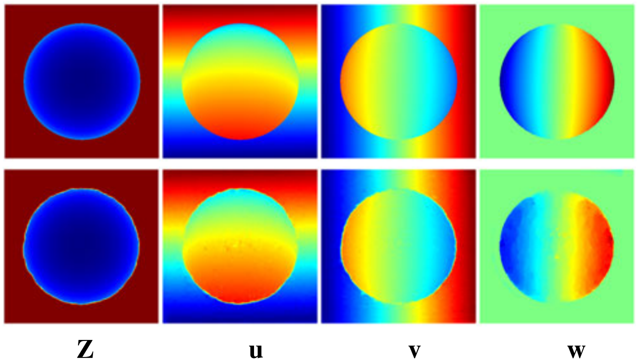

Figure 3. The top figure represents, from left to right, the ground truth for the depth Z and the 3D motion u, v and w. The bottom figure shows these results computed by our method.

Synthetic data: 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. 2). Therefore, occlusions and large discontinuities in both motion and depth must be dealt with. The accuracy of our results is demonstrated in Fig. 3 by comparing them with the ground truth depth and 3D motion. The results are quantitatively evaluated by computing the $NRMS_{\mathbf{P}}$, $NRMS_{\mathbf{V}}$ errors and the $AAE_{\mathbf{V}}$ (defined in Sec. 3). Table 2 summarizes the computed errors over three domains: all pixels, non-occluded regions, 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.

	$NRMS_{\mathbf{P}}$(%)	$NRMS_{\mathbf{V}}$(%)	$AAE_{\mathbf{V}}$ *(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

Table 2. The evaluated errors of our computed scene flow and structure over three domains: the continuous regions, the non-occluded regions and over all pixels.

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

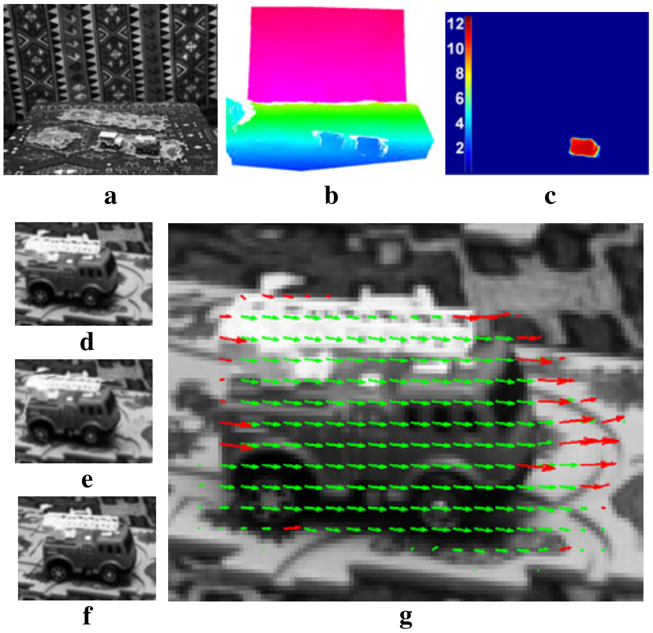

Figure 4. **Cars dataset:** **(a)**, the reference view at time t; **(b)**, the depth map masked with the computed occlusion maps; **(c)**, the magnitude of the computed scene flow (mm); **(d)**, zoom in at time t; **(e)**, the corresponding warped image; and **(f)**, zoom in at time $t+1$; **(g)**, the projection of the computed scene flow; Occluded pixels are colored in red.

of 1280 X 1024 and then were 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. 4) involves the rigid 3D motion of a small object (car), in a static scene. The second dataset (Fig. 5) 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. 6). In that case, the 3D motion is generally different for each 3D point. In addition the hair involves non-rigid motion. In all three datasets, large occlusions exist due to the noticeable dissimilarity between the frames.

We show our results in Fig. 4- 6. For each dataset we present: 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 colored in red. 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 recover the scene flow and depth. It can be observed that the warped images are very similar to the reference view.

5. 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 BC assumptions while enforcing geometric consistency between the views. The redundant information 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 non-convex optimization problem. Moreover, due to our 3D representation, the relation between the image coordinates and the unknowns is non-linear (as opposed to optical flow or disparity). Consequently, the derivation of the associated Euler-Lagrange equations involves non-trivial computations. In addition, the use of multiple views requires to properly handle occlusions since each view adds more occluded regions. Obviously, the occlusion between the views becomes more sever when a wide baseline rig is considered. Our variational framework, which is used for the first time

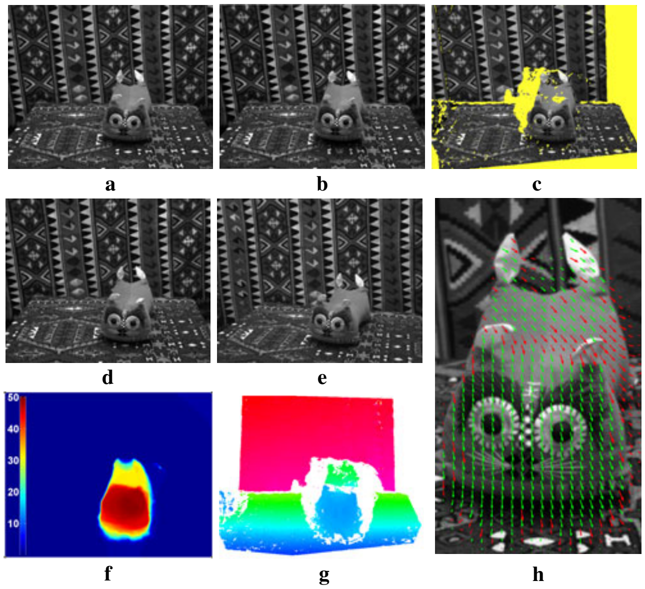

Figure 5. **Cat dataset:** **(a,d)**, the reference view at time t and $t+1$, respectively; **(e)**, the right view at time t; **(b,c)**, warped images from **d** $\rightarrow$ **a** and **e** $\rightarrow$ **a**, respectively; The yellow regions are the computed occlusions; **(f)**, the magnitude of the resulting scene flow (mm); **(g)**, the depth map masked with the computed occlusion maps; and **(h)**, the projection of the computed scene flow. Occluded pixels are colored in red.

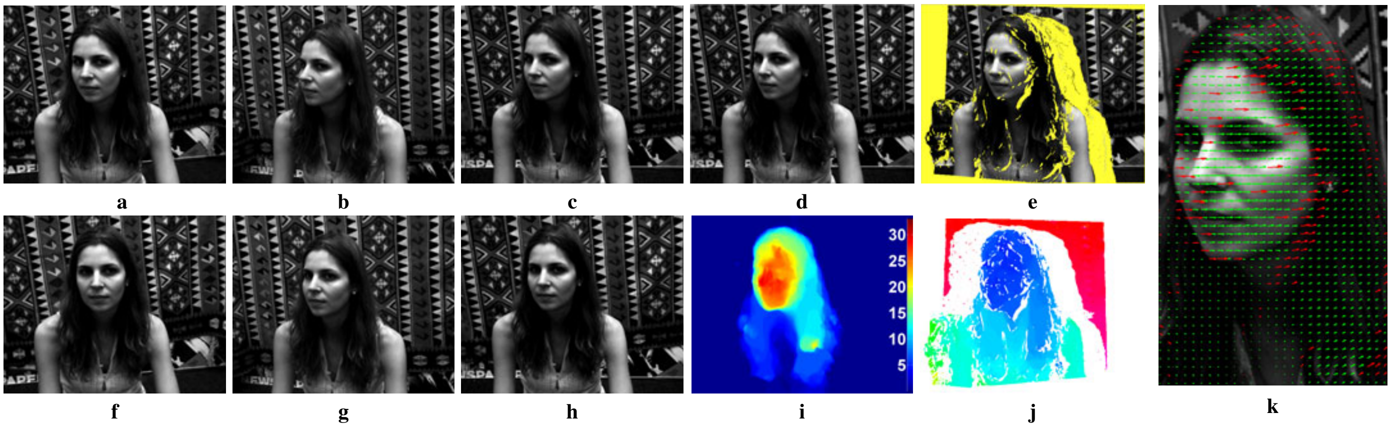

Figure 6. **Maria dataset**: **(a-c)**, the three views at time t, where **(c)** is the reference; **(f-h)**, the corresponding views at time $t+1$; **(d)**, warped image from **h**→ **c**; **(e)**, warped image from **f**→**c**, where the yellow regions are the computed occlusions; **(i)**, the magnitude 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.

for multiple views and 3D representation, successfully minimizes the resulting functional despite these difficulties.

Our accurate and dense results on real and synthetic data demonstrate the validity of the developed method. Most of the errors in our results are found in the depth discontinuities and in the occluded regions. These errors are expected to increase when the setup consists of even larger differences in the fields of view of the camera than those considered in our experiments. It is, therefore, worthwhile to further study a method that will better cope with such regions.

Acknowledgements

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

References

[1] T. Basha, Y. Moses, and K. N. Multi-View Scene Flow Estimation: A View Centered Variational Approach, TR, 2010. ftp://ftp.idc.ac.il/yael/papers/TR-BMK-2010.pdf.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 3024, pages 25–36, 2004.

[3] R. Carceroni and K. Kutulakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. *IJCV*, 49(2):175–214, 2002.

[4] J. Courchay, J. Pons, R. Keriven, and P. Monasse. Dense and accurate spatio-temporal multi-view stereovision. In *ACCV*, 2009.

[5] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[6] Y. Furukawa and J. Ponce. Dense 3D motion capture from synchronized video streams. In *CVPR*, pages 1–8, 2008.

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

[8] M. Isard and J. MacCormick. Dense motion and disparity estimation via loopy belief propagation. *ACCV*, 3852:32, 2006.

[9] R. Li and S. Sclaroff. Multi-scale 3D scene flow from binocular stereo sequences. *CVIU*, 110(1):75–90, 2008.

[10] D. Min and K. Sohn. Edge-preserving simultaneous joint motion-disparity estimation. In *ICPR*, volume 2, 2006.

[11] J. Neumann and Y. Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *IJCV*, 47(1):181–193, 2002.

[12] J. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV*, 72(2):179–193, 2007.

[13] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. *ECCV*, 1064:439–451, 1996.

[14] D. Scharstein and R. Szeliski. Middlebury stereo vision research page. http://vision.middlebury.edu/stereo.

[15] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

[16] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, volume 1, 2003.

[17] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV*, pages 722–729, 1999.

[18] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, pages 475–480, 2005.

[19] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6D. In *CVPR*, volume 2, 2000.

[20] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *ECCV*, 2008.

[21] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. *PAMI*, 31(12):2115, 2009.

[22] Y. Zhang and C. Kambhamettu. Integrated 3D scene flow and structure recovery from multiviewimage sequences. In *CVPR*, volume 2, 2000.

[23] Y. Zhang and R. Kambhamettu. On 3d scene flow and structure estimation. In *CVPR*, pages 778–785, 2001.