frontiers in ICT

ORIGINAL RESEARCH
published: 01 September 2015
doi: 10.3389/fict.2015.00016

Monocular concurrent recovery of structure and motion scene flow

Amar Mitiche[1]*, *Yosra Mathlouthi*[1] *and Ismail Ben Ayed*[2]†

[1] Institut National de la Recherche Scientifique (INRS-EMT), Montréal, QC, Canada, [2] GE Healthcare, London, ON, Canada

This paper describes a variational method of joint three-dimensional structure and motion scene flow recovery from a single image sequence. A basic scheme is developed by minimizing a functional with a term of conformity of scene flow and depth to the image sequence spatiotemporal variations, and quadratic smoothness regularization terms. The data term follows by rewriting optical velocity in the optical flow gradient constraint in terms of scene flow and depth. As a result, this problem statement is analogous to the classical Horn and Schunck optical flow formulation except that it involves scene flow and depth rather than image motion. When discretized, the Euler–Lagrange equations give a large scale sparse system of linear equations in the unknowns of the scene flow three coordinates and depth. The equations can be ordered in such a way that the corresponding matrix is symmetric positive definite, so that they can be solved efficiently by Gauss–Seidel iterations. Experiments are shown to verify the scheme's validity and efficiency.

Keywords: computer vision, motion analysis, scene flow, 3D motion, depth

OPEN ACCESS

Edited by:
Marcello Pelillo,
Ca' Foscari University of Venice, Italy

Reviewed by:
Carlos Vazquez,
École de Technologie Supérieure, Canada
Jing Yuan,
Western University, Canada
Antonio Fernández-Caballero,
Universidad de Castilla-La Mancha, Spain

****Correspondence:***
Amar Mitiche,
Institut National de la Recherche Scientifique (INRS-EMT), 800, de la Gauchetière Ouest, Montréal, QC H5A 1K6, Canada
mitiche@emt.inrs.ca

†Present address:
Ismail Ben Ayed,
École de Technologie Supérieure, Montréal, QC, Canada

Specialty section:
This article was submitted to Computer Image Analysis, a section of the journal Frontiers in ICT

Received: *18 February 2015*
Accepted: *10 August 2015*
Published: *01 September 2015*

Citation:
Mitiche A, Mathlouthi Y and Ben Ayed I (2015) Monocular concurrent recovery of structure and motion scene flow. Front. ICT 2:16. doi: 10.3389/fict.2015.00016

1. Introduction

Scene flow is the field over the image domain of the visible environmental surfaces three-dimensional (3D) velocities. Only the visible surfaces are relevant in the definition because they alone, not the occluded, contribute to visual information. As a result, the scene flow domain is the image domain: at each image point, scene flow consists of the velocity vector of the corresponding visible environmental surface point. It is the time derivative of the point 3D position.

For a working definition of scene flow, let Ω be the common domain of an image sequence $I(\mathbf{x}, t)$, where $\mathbf{x} = (x, y)$ designates image position, and t is time. For each point, $\mathbf{x} \in \Omega$, let $\mathbf{P}' = \frac{d\mathbf{P}}{dt}$ be the velocity vector of the visible environmental point $\mathbf{P} = (X, Y, Z)$ projected on $\mathbf{x}$. Scene flow is the velocity vector field, $\mathbf{F} = (U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ over Ω. It is a function of image position and time: $\mathbf{F} = \mathbf{F}(\mathbf{x}, t)$.

Scene flow is a fundamental dimension of three-dimensional scene analysis for the obvious reason that it describes the motion of real objects in the environment. Moreover, it is related to optical flow via depth (Mitiche and Aggarwal, 2013). It can be regarded as a three-dimensional analog of *optical flow*: at each point $\mathbf{x} \in \Omega$, and each instant of time, scene flow is the velocity vector of the visible environmental point $\mathbf{P}$, which projects on $\mathbf{x}$, whereas optical flow is the velocity vector of the image of $\mathbf{P}$ at $\mathbf{x}$.

From a broad perspective, scene flow can be computed in one of two ways: *parametric* and *non-parametric*. Parametric methods use a parametric form of the scene flow coordinates and non-parametric methods compute scene flow directly as a vector field without resorting to an intermediate representation.

Investigations of parametric scene flow generally assume that environmental objects are rigid and, therefore, decompose scene flow in terms of 3D translational and rotational parameters (Mitiche and Aggarwal, 2013). This representation leads to the Longuet–Higgins and Pradzny fundamental equations (Longuet-Higgins and Prazdny, 1980) relating the rigid motion parameters, depth, and optical flow. Depth and the rigid screw motion parameters become the unknown 3D variables to determine, from which scene flow can be recovered *a posteriori*. In most studies, the Longuet–Higgins and Pradzny equations underlie the recovery of rigid body structure and motion from image sequences, even when not used explicitly.

Current parametric methods can be separated into two broad categories, those which treat the case of a viewing system moving in a static environment and those which allow the viewing system and the environmental objects to move simultaneously and independently. In the first case, the problem is significantly simpler because the single 3D motion to take into account is that of the viewing system (Bruss and Horn, 1983; Adiv, 1985; Horn and Weldon, 1988; Shahraray and Brown, 1988; Heeger and Jepson, 1992; Taalebinezhaad, 1992; De Micheli and Giachero, 1994; Gupta and Kanal, 1995; Xiong and Shafer, 1995; Hung and Ho, 1999; Brodsky et al., 2000; Srinivasan, 2000; Liu et al., 2002). Moreover, segmentation of the environment into differently moving objects is not an issue in a static environment, simplifying the problem further.

The simultaneous motion of the viewing system and viewed objects has also been the subject of several studies (MacLean et al., 1994; Weber and Malik, 1997; Fejes and Davis, 1998; Mitiche and Hadjres, 2003; Mitiche and Sekkati, 2006; Sekkati and Mitiche, 2006a,b, 2007). The non-variational methods in MacLean et al. (1994), Weber and Malik (1997), and Fejes and Davis (1998) assume that optical flow is given beforehand and segment the visual field into differently moving rigid objects by grouping processes, such as region growing by 3D motion (Weber and Malik, 1997), clustering of 3D motion via mixture models (MacLean et al., 1994), and clustering via oriented projections of optical flow (Fejes and Davis, 1998).

The variational methods in Mitiche and Hadjres (2003), Mitiche and Sekkati (2006), and Sekkati and Mitiche (2006a,b, 2007) use functionals with a data term based on the Longuet–Higgins and Prazdny rigid motion model and a regularization term to account for 3D interpretation discontinuities; they mainly differ in the way these discontinuities are represented.

Non-parametric scene flow computation methods seek to recover scene flow at each point of the image domain without recourse to a parametric form of the movements or surfaces in space. Such methods are most relevant when practicable models of scene flow cannot be assumed, as with, for instance, articulated human and animal motion.

Because scene flow is related to depth, via optical flow, which they jointly define (Mitiche and Aggarwal, 2013), non-parametric scene flow computation has been generally studied in the context of stereoscopy (Zhang and Kambhamettu, 2000; Pons et al., 2003; Vedula et al., 2005; Huguet and Devernay, 2007; Wedel et al., 2008, 2011; Rabe et al., 2010; Basha et al., 2013; Vogel et al., 2013), although it stands independent of stereoscopy. Here following, we will show that non-parametric scene flow can actually be recovered from a single image sequence. We will describe a variational scheme reminiscent of the Horn and Schunck optical flow estimation method. The functional of this formulation has two terms: a data term, which relates 3D velocity to depth via the image sequence spatiotemporal variations, and a classic smoothness regularization term. The data term falls out simply by rewriting the Horn and Schunck optical flow constraint linearly in terms of scene flow and depth. The Euler–Lagrange equations corresponding to the minimization of the objective functional yield, when discretized, a large sparse system of linear equations, which can be solved efficiently by Jacobi/Gauss–Seidel iterations. The scheme can be generalized to boundary preserving formulations as in optical flow estimation (Deriche et al., 1995; Aubert et al., 1999).

The remainder of this paper is organized as follows: Section 2 formulates the problem and develops the objective functional. Section 3 deals with the optimization of the objective functional. It derives the Euler–Lagrange equations and the corresponding discrete system of linear equations in the variables of scene flow and relative depth. It also shows that the matrix of this system is symmetric positive definite, which prescribes a solution by Jacobi/Gauss–Seidel iterations. Section 4 addresses the problem of regularized spatiotemporal derivative computation and Section 5 gives experimental results.

2. Formulation

The problem is to recover scene flow and depth from an image sequence I: $(x, y, t) \rightarrow I(x, y, t)$, where (x, y) are the coordinates over the bounded image domain Ω, and $t \in \mathbb{R}^+$ is time. The formulation starts with the Horn and Schunck optical flow gradient constraint (Horn and Schunck, 1981), which relates the coordinate functions, u and v, of optical flow to the image sequence spatiotemporal variations:

$$I_x u + I_y v + I_t = 0, \tag{1}$$

where I_x, I_y, I_t are the image spatiotemporal partial derivatives. Let **P** be a point in space (X, Y, Z), its 3D coordinates, and (x, y) its image coordinates. The viewing system model geometry is shown in **Figure 1**. Derivation with respect to time of the projection equations $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$, where f is the focal length, gives the coordinates u, v of optical velocity as functions of scene flow and depth:

$$u = \frac{dx}{dt} = \frac{fU - xW}{Z}; \quad v = \frac{dy}{dt} = \frac{fV - yW}{Z}, \tag{2}$$

where Z designates depth (**Figure 1**) and $(U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ is the scene flow at **P**. Substitution of these optical flow expressions in the gradient constraint (1), followed by the multiplication of the left hand side by $Z \neq 0$ gives the following linear constraint relating scene flow and depth to the image spatiotemporal derivatives:

$$fI_x U + fI_y V - (xI_x + yI_y)W + I_t Z = 0. \tag{3}$$

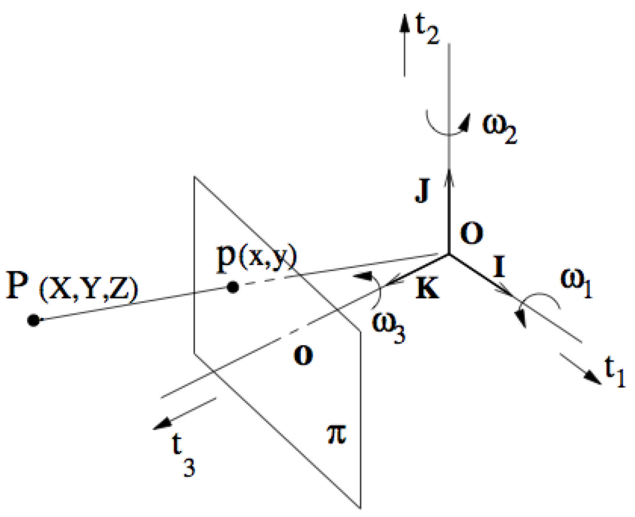

FIGURE 1 | The viewing system is symbolized by a Cartesian reference system (O; *X,Y,Z*) and central projection through the origin. The *Z*-axis is the depth axis. The image plane π is orthogonal to the *Z*-axis at distance *f*, the focal length, from **O**.

There are two important observations to make about this linear equation. First, an obvious observation is that this equation evaluates at each point of the image domain: $fI_x(\mathbf{x})U(\mathbf{x}) + fI_y(\mathbf{x})\,V(\mathbf{x}) - (xI_x(\mathbf{x}) + yI_y(\mathbf{x}))W(\mathbf{x}) + I_t(\mathbf{x})\mathrm{Z}(\mathbf{x}) = 0$ and, therefore, contains four unknown variables at each point. As with optical flow computation (Hildreth, 1984), this says that any local interpretation of the variables is ambiguous. Dense interpretation, i.e., global over the image domain, which is of interest to us here, will require additional constraints. In a classic way, we will use a variational statement of the problem where these additional constraints characterize the variables as smooth over the image domain.

The second observation is that the equation, being homogeneous, has a trivial solution, $U = V = W = 0$; $Z = 0$, in which, of course, we are not interested, and can easily avoid in practice. More importantly, since we are dealing with an actual physical problem, the environmental depth field and scene flow, which gave rise to the image spatiotemporal variations, constitute a solution but so does any scaled version of it: if (U, V, W), Z is a solution, so is $k(U, V, W)$, kZ for some arbitrary real k. This is a limitation inherent to the recovery of 3D structure and motion from 2D image sequences (Ullman, 1983).

Theoretically, the scale can be fixed by setting the depth of a particular point, in which case the depth of any other point is relative to it. However, this is hardly an answer, because setting the depth of a point would affect a single equation out of thousands that generally make up the discrete system of equations in practice. Along a different vein, a particular solution can be picked by imposing it a given norm. For instance, we could, say, compute a unit norm solution: $\|(U, V, W, Z)\| = 1$ by solving under this constraint the system of equations of the problem. We will follow instead an effective simple scheme: we will adopt a variational formulation of the problem where the scale of interpretation is fixed by solving for depth Z_r relative to a fronto-parallel plane $\Pi_{Z_0}: Z = Z_0$, for some arbitrary positive depth, Z_0. More precisely, this is done by first rewriting Eq. (3) as follows:

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_t(Z - Z_0) + I_tZ_0 = 0, \tag{4}$$

and then making a change of variable $Z_r \leftarrow Z - Z_0$, which would give:

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ_r + I_tZ_0 = 0 \tag{5}$$

For notational simplicity and economy, we can reuse the symbol Z to designate relative depth Z_r, in which case we write Eq. (5) as:

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0 \tag{6}$$

By rewriting this data equation with respect to reference, plane Π_{Z_0} effectively fixes the scale of 3D interpretation because for another reference plane, say $Z_1 = kZ_0$, the equation integrity is maintained by the correspondingly scaled interpretation $k(U, V, W)$, kZ and, inversely, for a scaled solution $k(U, V, W)$, kZ the equation integrity is maintained by a reference plane at kZ_0. The trivial solution, which is now $(0, 0, 0, -Z_0)$, rather than $(0, 0, 0, 0)$, can be avoided in practice simply by initializing away from it in the iterative algorithm that we are about to develop. This iterative algorithm, as will be detailed subsequently, consists of Gauss–Seidel iterations, which at each step solve by singular value decomposition local 4×4 systems of linear equations resulting from the objective functional Euler–Lagrange equations.

We can now formulate the problem of joint computation of scene flow and relative depth from a single image sequence as the minimization of the following functional:

$$\begin{aligned}\mathbf{E}(U, V, W, Z|I) = {} & \frac{1}{2}\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ \\ & + I_tZ_0)^2 dxdy \\ & + \frac{\alpha}{2}\int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2)dxdy \\ & + \frac{\beta}{2}\int_\Omega \|\nabla Z\|^2 dxdy,\end{aligned} \tag{7}$$

where α and β are positive constants balancing the contributions of the smoothness terms. This functional can be modified to preserve the boundaries of scene flow/depth via discontinuity preserving regularization (Deriche et al., 1995).

3. Optimization

The Euler–Lagrange equations corresponding to the objective functional (7) are the following coupled partial differential equations:

$$\begin{aligned}& fI_x(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha\nabla^2 U = 0 \\ & fI_y(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha\nabla^2 V = 0 \\ & (-xI_x - yI_y)(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & \quad - \alpha\nabla^2 W = 0 \\ & I_t(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \beta\nabla^2 Z = 0,\end{aligned} \tag{8}$$

to which we add the Neumann boundary conditions on the solution at the boundary $\partial\Omega$ of Ω:

$$\frac{\partial U}{\partial \mathbf{n}} = 0, \quad \frac{\partial V}{\partial \mathbf{n}} = 0, \quad \frac{\partial W}{\partial \mathbf{n}} = 0, \quad \frac{\partial Z}{\partial \mathbf{n}} = 0, \tag{9}$$

where $\frac{\partial}{\partial \mathbf{n}}$ is the differentiation operator in the direction of the normal $\mathbf{n}$ of $\partial\Omega$.

Let Ω be discretized as a unit-spacing grid D and the grid points indexed by the integers $\{1,2, \ldots, N\}$. Pixels are indexed in the lexicographical order, i.e., top-down and left-to-right. $N = n^2$ when the image is $n \times n$. Let $a = fI_x$, $b = fI_y$, $c = -(xI_x + yI_y)$, $d = I_t$. For all grid point $I \in \{1, 2, \ldots, N\}$, a discrete approximation of the Euler–Lagrange equations (8) is:

$$\begin{aligned}
&a_i^2 U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i + a_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (U_j - U_i) = 0 \\
&b_i a_i U_i + b_i^2 V_i + b_i c_i W_i + b_i d_i Z_i + b_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (V_j - V_i) = 0 \\
&c_i a_i U_i + c_i b_i V_i + c_i^2 W_i + c_i d_i Z_i + c_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (W_j - W_i) = 0 \\
&d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + d_i^2 Z_i + d_i^2 Z_0 - \beta \sum_{j \in \mathbf{N}_i} (Z_j - Z_i) = 0
\end{aligned} \tag{10}$$

where $(U_{i,}\ V_i,\ W_i,\ Z_i) = (U,\ V,\ W,\ Z)_i$ is the scene flow at i; a_i, b_i, c_i, d_i are the values at i of a, b, c, d, respectively, and $\mathbf{N}_i$ is the set of indices of the neighbors of i. For the four-neighborhood, *card* $(\mathbf{N}_i) = 4$ for points interior in D, and *card* $(\mathbf{N}_i) < 4$ for boundary points. The Laplacian $\bigtriangledown^2 Q$, $Q \in \{U,\ V,\ W,\ Z\}$, in the Euler–Lagrange equations, has been discretized as $\frac{1}{4}\sum_{j \in \mathbf{N}_i} (Q_j - Q_i)$, where the factor $\frac{1}{4}$ is absorbed by α and β. Rewriting (10), and where $n_i = card(\mathbf{N}_i)$, we have the following system of linear equations, $I \in \{1, \ldots, N\}$:

$$(S)\begin{cases}
(a_i^2 + \alpha n_i) U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i - \alpha \sum\limits_{j \in \mathbf{N}_i} U_j = -a_i d_i Z_0 \\
b_i a_i U_i + (b_i^2 + \alpha n_i) V_i + b_i c_i W_i + b_i d_i Z_i - \alpha \sum\limits_{j \in \mathbf{N}_i} V_j = -b_i d_i Z_0 \\
c_i a_i U_i + c_i b_i V_i + (c_i^2 + \alpha n_i) W_i + c_i d_i Z_i - \alpha \sum\limits_{j \in \mathbf{N}_i} W_j = -s c_i d_i Z_0 \\
d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + (d_i^2 + \beta n_i) Z_i - \beta \sum\limits_{j \in \mathbf{N}_i} Z_j = -d_i^2 Z_0
\end{cases}$$

Let $\mathbf{q} = (q_1,\ \ldots,\ q_{4N})\ ^t \in\ \mathbb{R}^{4N}$ be the vector with coordinates $q_{4i-3} = U_i$, $q_{4i-2} = V_i$, $q_{4i-1} = W_i, q_{4i} = Z_i$, $I \in \{1,\ \ldots,\ N\}$, and $\mathbf{r} = (r_1, \ldots r_{4N})^t \in \mathbf{R}^{4N}$, the vector with coordinates $r_{4i-3} = -a_i\ d_i Z_{0,}\ r_{4i-2} = -b_i\ d_i Z_0$, $r_{4i-1} = -c_i\ d_i Z_0$, and $r_{4i}\ =\ -d_i^2 Z_0$, $i \in \{1, \ldots, N\}$. System (S) of linear equations can be written in matrix form as:

$$\mathbf{Aq} = \mathbf{r} \tag{11}$$

where $\mathbf{A}$ is the $4N \times 4N$ matrix with elements $\mathbf{A}_{4i-3,4i-3} = a_i^2 + \alpha n_i$; $\mathbf{A}_{4i-2,4i-2}\ =\ b_i^2\ +\ \alpha n_i$; $\mathbf{A}_{4i-1,4i-1}\ =\ c_i^2\ +\ \alpha n_i$; $\mathbf{A}_{4i,4i}\ = d_i^2 + \beta n_i$; $\mathbf{A}_{4i-3,4i-2} = \mathbf{A}_{4i-2,4i-3} = a_i b_i$; $\mathbf{A}_{4i-3,4i-1} = \mathbf{A}_{4i-1,4i-3} = a_i c_i$; $\mathbf{A}_{4i-3,4i} = \mathbf{A}_{4i,4i-3} = a_i d_i$; $\mathbf{A}_{4i-2,4i-1} = \mathbf{A}_{4i-1,4i-2} = b_i c_i$; $\mathbf{A}_{4i-2,4i} = \mathbf{A}_{4i,4i-2} = b_i d_i$; $\mathbf{A}_{4i-1,4i} = \mathbf{A}_{4i,4i-1} = c_i d_i$; for all $\mathrm{I} \in \{1,\ \ldots,\ \mathrm{N}\}$; $\mathbf{A}_{4i-3,4j-3} = \mathbf{A}_{4i-2,4j-2} = \mathbf{A}_{4i-1,4j-1} = -\alpha$ and $\mathbf{A}_{4i-4j} = -\beta$, for all $i,\ j \in \{1, \ldots, N\}$ such that $j \in \mathrm{N}_i$, all other elements being equal to zero.

System (S) is a large scale sparse system of linear equations. Such systems are best solved by iterative methods designed for sparse matrices (Ciarlet, 1982; Stoer and Bulirsch, 2002). Here following, we prove that matrix $\mathbf{A}$ is symmetric positive definite, which implies an effective solution of Eq. (11) by 4×4 block-wise Gauss–Seidel iterations.

One can easily verify that matrix $\mathbf{A}$ is symmetric. Matrix $\mathbf{A}$ is also positive definite. To show this, we verify that $\mathbf{q}^t Aq > 0$ for all $\mathbf{q} \in \mathbf{R}^{4N}$, $\mathbf{q} \neq 0$. We have:

$$\begin{aligned}
&\mathbf{q}^t\mathbf{Aq} \\
&= \sum_{i=1}^{N} \left((a_i^2 + \alpha n_i) U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} U_j \right) U_i \\
&+ \sum_{i=1}^{N} \left(b_i a_i U_i + (b_i^2 + \alpha n_i) V_i + b_i c_i W_i + b_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} V_j \right) V_i \\
&+ \sum_{i=1}^{N} \left(c_i a_i U_i + c_i b_i V_i + (c_i^2 + \alpha n_i) W_i + c_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} W_j \right) W_i \\
&+ \sum_{i=1}^{N} \left(d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + (d_i^2 + \beta n_i) Z_i - \beta \sum_{j \in \mathbf{N}_i} Z_j \right) Z_i
\end{aligned} \tag{12}$$

Following algebraic manipulations, we get:

$$\begin{aligned}
\mathbf{q}^t\mathbf{Aq} = &\sum_{i=1}^{N} (a_i U_i + b_i V_i + c_i W_i + d_i Z_i)^2 \\
&+ \alpha \sum_{i=1}^{N} \left(n_i (U_i^2 + V_i^2 + W_i^2) \right) + \beta \sum_{i=1}^{N} \left(n_i (Z_i^2) \right) \\
&- \alpha \sum_{i=1}^{N} \left(\sum_{j \in \mathbf{N}_i} U_j U_i + \sum_{j \in \mathbf{N}_i} V_j V_i + \sum_{j \in \mathbf{N}_i} W_j W_i \right) \\
&- \beta \sum_{i=1}^{N} \left(\sum_{j \in \mathbf{N}_i} Z_j Z_i \right)
\end{aligned} \tag{13}$$

If we distribute the n_i terms U_i of the second row into the corresponding neighborhood sum of the third row, we will get:

$$\begin{aligned}
\sum_{i=1}^{N} n_i U_i^2 - \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i} U_j U_i &= \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j > i} (U_i^2 + U_j^2 - 2 U_j U_i) \\
&= \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j > i} (U_i - U_j)^2
\end{aligned}$$

Using similar manipulations for the other variables (V,W,Z), we arrive at the expression we need:

$$\mathbf{q}^t\mathbf{A}\mathbf{q} = \sum_{i=1}^{N} (a_iU_i + b_iV_i + c_iW_i + d_iZ_i)^2 + \alpha\sum_{i=1}^{N}\sum_{j\in\mathbf{N}_i;j>i}\left((U_i - U_j)^2 + (V_i - V_j)^2 + (W_i - W_j)^2\right) + \beta\sum_{i=1}^{N}\sum_{j\in\mathbf{N}_i;j>i}\left((Z_i - Z_j)^2\right) \quad (14)$$

For $\mathbf{q} \neq 0$, we have $\mathbf{q}^t\ \mathbf{A}\mathbf{q} = 0$ if and only if the terms in both sums on the right-hand side of (14) are zero. The second-sum terms are zero if and only if the scene consists of a fronto-parallel plane (plane $Z = Z_0$) under constant translation ($(U_i,\ V_i,\ W_i) = \mathbf{T}$). The first-sum terms are zero if and only if all vectors $(a_i,b_i,c_i,\mathrm{d}_i)_i = (I_{xi},I_{yi},\ -x_iI_{xi} - y_iI_{yi,}\ I_{ti})$ lie in a hyperplane for all $(x_i,y_i) \in D$. This is possible if and only if the spatiotemporal visual pattern is null, which is an irrelevant case. Therefore, $\mathbf{q}^t\ \mathbf{A}\mathbf{q} > 0$ for $\mathbf{q} \neq 0$ and $\mathbf{A}$ is positive definite. This means that the point-wise and block-wise Gauss–Seidel and relaxation iterative methods for solving system (11) converge (Ciarlet, 1982; Stoer and Bulirsch, 2002).

For a 4 × 4 block division of matrix $\mathbf{A}$, the Gauss–Seidel iterations consist of solving, for each i, $\in\{1, \ldots, N\}$, the following 4 × 4 linear system of equations, where k is the iteration number:

$$(a_i^2 + \alpha n_i)U_i^{k+1} + a_ib_iV_i^{k+1} + a_ic_iW_i^{k+1} + a_id_iZ_i^{k+1} = -a_id_iZ_0 + \alpha\left(\sum_{j\in\mathbf{N}_i;j<i} U_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} U_j^k\right)$$

$$b_ia_iU_i^{k+1} + (b_i^2 + \alpha n_i)V_i^{k+1} + b_ic_iW_i^{k+1} + b_id_iZ_i^{k+1} = -b_id_iZ_0 + \alpha\left(\sum_{j\in\mathbf{N}_i;j<i} V_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} V_j^k\right)$$

$$c_ia_iU_i^{k+1} + c_ib_iV_i^{k+1} + (c_i^2 + \alpha n_i)W_i^{k+1} + c_id_iZ_i^{k+1} = -c_id_iZ_0 + \alpha\left(\sum_{j\in\mathbf{N}_i;j<i} W_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} W_j^k\right)$$

$$d_ia_iU_i^{k+1} + d_ib_iV_i^{k+1} + d_ic_iW_i^{k+1} + (d_i^2 + \beta n_i)Z_i^{k+1} = -d_i^2Z_0 + \beta\left(\sum_{j\in\mathbf{N}_i;j<i} Z_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} Z_j^k\right),$$

which can be done efficiently by the singular value decomposition method (Forsythe et al., 1977).

4. Estimation of the Spatiotemporal Derivatives

The purpose is to estimate the spatiotemporal derivatives I_x, I_y, I_t, from two consecutive images of a sequence. The estimation of a function derivative from inaccurate data is an ill-posed problem because small changes in the function values can result in arbitrarily large errors in the derivative estimated by finite differences (Terzopoulos, 1986). Therefore, image noise can adversely affect the quality of motion interpretation that uses finite difference image derivatives. In motion analysis, the problem has been generally approached by local averaging of the finite difference derivatives (Horn and Schunck, 1981). However, regularized differentiation can be more effective as we show in the following.

4.1. Differentiation by Averaging Finite Differences

Following the formulas in the Horn and Schunck paper on optical estimation (Horn and Schunck, 1981), motion analysis studies have generally used forward first differences to represent derivatives, locally averaged to counter the effect of noise:

$$\begin{aligned} I_x(r,c) &\approx \tfrac{1}{4}\sum_{\Delta r=0}^{1} \{\, I(r+\Delta r, c+1, 0) - I(r+\Delta r, c, 0) + I(r+\Delta r, c+1, 1) - I(r+\Delta r, c, 1)\,\} \\ I_y(r,c) &\approx \tfrac{1}{4}\sum_{\Delta c=0}^{1} \{\, I(r+1, c+\Delta c, 0) - I(r, c+\Delta c, 0) + I(r+1, c+\Delta c, 1) - I(r, c+\Delta c, 1)\,\} \\ I_t(r,c) &\approx \tfrac{1}{4}\sum_{\Delta r=0}^{1}\sum_{\Delta c=0}^{1} \{I(r+\Delta r, c+\Delta c, 1) - I(r+\Delta r, c+\Delta c, 0)\} \end{aligned} \quad (15)$$

where I_0 is the current image and I_1 the next. The spatial derivatives have sometimes been estimated using averages of central differences.

Global averaging of the finite difference approximations can be done using L^2 smoothing of the derivatives finite differences: a derivative estimate g is computed by minimizing:

$$E(g) = \frac{1}{2}\int_{\Omega} \left((g - g_0)^2 + \gamma \parallel \nabla g \|^2\right) dxdy, \quad (16)$$

where g is a partial derivative function and g_0 its finite difference approximation from I. The corresponding Euler–Lagrange equation $g - g_0 - \gamma\nabla^2 g = 0$ is then discretized to yield a large sparse system of linear of equations.

4.2. Regularized Differentiation

Although image data smoothing is commonly done in motion analysis, it does not generally solve the derivative estimation ill-posedness, so that prior de-noising of the image independently of differentiation followed by finite difference approximation is not generally effective (Chartrand, 2005). A more productive method is to state differentiation within Tikhonov regularization theory for ill-posed problems (Cullum, 1971; Hanke and Scherzer, 2001). In Hanke and Scherzer (2001), the problem was to find a smooth approximation of the true derivative y' of a function y from given data $\tilde{y}_i$. This was done by determining an approximation f of y, which minimizes an objective functional having a term of discrepancy between f and the given data, and a regularization term to penalize the L^2 norm of f''. The derivative was subsequently evaluated on f. The objective functional in Hanke and Scherzer (2001) was investigated in earlier studies (Schoenberg, 1964; Reinsch, 1967), which showed that it is minimized by a natural cubic spline. In Cullum (1971), the differentiation process itself was regularized: the formulation sought to determine an approximation u of the true derivative, which minimized a functional containing a data fidelity term via a Fredholm integral of anti-differentiation, and penalty term via the L^2 norm of u and u'. More recently, Chartrand (2005) investigated total variation (TV) regularization in conjunction with an anti-differentiation data discrepancy term as in Cullum (1971). The discrete implementation of the ensuing problem followed a standard numerical scheme in TV restoration (Vogel, 2002).

In the following, we will estimate the derivatives, I_x and I_y, by a variational method, which uses an anti-differentiation data discrepancy term as in Cullum (1971) and Chartrand (2005) and an L^2 smoothness regularization. Enforcing smoothness on the derivatives is consistent with the L^2 regularization in the scene flow estimation scheme we have described. For reasons that will become clearer later, the formulation does not apply to I_t given that the time axis is sampled only at two points; recall that we are to estimate the derivatives from two consecutive images. Instead, I_t can be estimated by regularized forward differences or simply by the Horn and Schunck formulas.

We will describe the method for I_x. The derivative I_y can be treated by the same formulas using the transposed image. Consider I_x at some fixed time t, so that it is viewed as a function of the image spatial coordinates but not of time. For convenience, we will also drop time from the coordinates of I. Let $\Omega = [0,l] \times [0,l]$. The partial derivative, I_x, will be computed as the minimizer of the following functional:

$$E(g) = \frac{1}{2} \int_{\Omega} \left(\| Ag - I \|^2 + \lambda \| \nabla g \|^2 \right) dxdy \tag{17}$$

where $\bigtriangledown$ is the spatial gradient, λ is a positive constant, and A is the integral operator of anti-differentiation defined by:

$$Ag(x,y) = \int_0^x g(z,y)dz \tag{18}$$

The Euler–Lagrange equation corresponding to (18) is:

$$A^*(Ag - I) - \lambda \nabla^2 g = 0 \tag{19}$$

where A^* is the adjoint operator of A, defined by:

$$A^* g(x,y) = \int_x^l g(z,y)dz \tag{20}$$

Here following is a discretization of Eq. (19) leading to a large-scale sparse system of linear equations. As before, let the points of the discretization grid D be listed top-down and left to right. The image in this lexicographical order is $I \in \mathbf{R}^N$, where $N = n^2$ for an image of size $n \times n$. Let $g_i, i = 1, \ldots, N$, be g evaluated at grid point I, and $\mathbf{g} \in \mathbf{R}^N$ the corresponding vector. For simplicity, we will use the same symbol to designate the linear operators, A and A^*, in (19) and their corresponding discretization matrix. Using the composite trapezoid quadrature rule for integral approximation (with one-pixel data spacing) (Forsythe et al., 1977), the $N \times N$ matrix A is defined by:

$$\begin{gathered} A(kn+i, kn+1) = \tfrac{1}{2};\ i = 2, \ldots, n;\ k = 0, \ldots, n-1 \\ A(kn+i, kn+i) = \tfrac{1}{2};\ i = 2, \ldots, n;\ k = 0, \ldots, n-1 \\ A(kn+i, kn+i-j) = 1; \quad i = 3, \ldots, n;\ j = 1, \ldots, i-2, \\ k = 0, \ldots, n-1, \end{gathered}$$

and all of the other elements are zero. The elements of rows $kn + 1$; $k = 0, \ldots, n - 1$ are zero to reflect the integral in (18) when $x = 0$. Matrix A is block diagonal sparse, with blocks of size $n \times n$. The $N \times N$ matrix A^* is similarly defined:

$$\begin{gathered} A^*(i,i) = \tfrac{1}{2};\ i \in [1, n^2],\ i \neq kn,\ k = 1, \ldots, n \\ A^*(kn+i, (k+1)n) = \tfrac{1}{2};\ i = 1, \ldots, n-1;\ k = 0, \ldots, n-1 \\ A^*(kn+i, kn+i+j) = 1;\ i = 1, \ldots, n-1; \\ j = 1, \ldots, n-i-1,\ k = 0, \ldots, n-1, \end{gathered}$$

and all the other elements are zero. The elements of rows kn, $k = 1, \ldots, n$ are zero to reflect the integral in (21) when $x = l$. Matrix A^* is block diagonal sparse, with blocks of size $n \times n$. The Laplacian term in Eq. (20) can be discretized as $\lambda \sum_{j \in \mathbf{N}_i} (g_j - g_i)$, where the factor of the approximation is absorbed by β, and $\mathbf{N}_i$ is the set of indices of the neighbors of i. The corresponding matrix is defined by:

$$\begin{gathered} L(i,i) = -\lambda n_i; \quad i = 1, \ldots, N \\ L(i,j) = \lambda; \quad j \in \mathbf{N}_i, \end{gathered}$$

where $n_i = card(\mathbf{N}_i)$. The system of linear equations to solve is:

$$\left(A^* A - L \right) \mathbf{g} = A^* I \tag{21}$$

This large scale sparse system of linear equations can be solved efficiently by an iterative method such Gauss–Seidel.

4.3. Example

Here following is an example. It uses the noised synthetic pyramidal image of **Figure 2**. The derivatives computed using regularized differentiation and the Horn and Schunck averaging are shown graphically in the figure. The values computed by regularized differentiation are closer to the true values: the mean squared error between true and computed values are 0.0409 for the regularized values and 1.086 for the Horn and Schunck averaging. Derivatives are measured in gray levels [(0 255) range] per pixel. The value of λ is 5.0 and the SNR is 0.5.

5. Experimental Results

This section presents various experiments on synthetic and real sequences to verify the validity of the method and its implementation. We show the recovered depth using anaglyphs (red/cyan) and color-coded displays, and novel viewpoint images. Color-coded depth is a standard display style. Anaglyphs are a convenient means for the subjective appraisal of the computed object structure. They are constructed from one of the two input images used in the experiment and the recovered depth map. Anaglyphs are best perceived on good-quality photographic paper. When

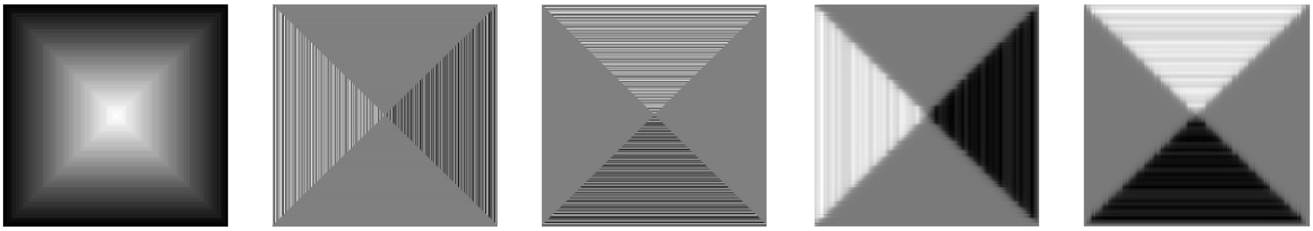

FIGURE 2 | From the left to the right: the noised 2D pyramidal image (SNR = 0.5); the partial derivatives I_x and I_y using Horn and Schunck averaging of forward image differencing; the partial derivatives, I_x and I_y, using regularized differencing (λ = 5.0).

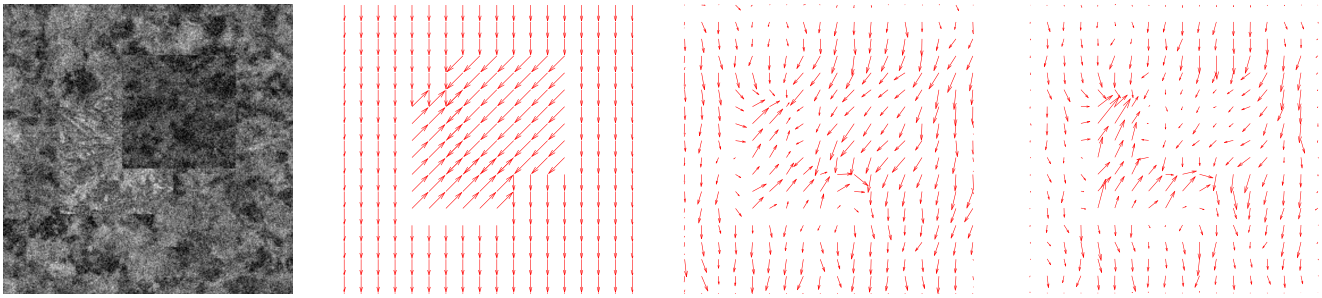

FIGURE 3 | *Synthetic squares* sequence. From left to right: The first of the two images; the vector-coded ground truth; optical flow corresponding to the estimated scene flow; optical flow computed directly by the Horn and Schunck method.

viewed on standard screens, they are generally better perceived with full color resolution. Finally, we also show a novel viewpoint image, i.e., a picture of the reconstructed object as viewed from a viewpoint different from the one of either of the two input images.

For scene flow, we show a vector display of its projection from some viewpoint. Also, and since we have no ground truth of scene flow for the used sequences, we show the optical flow corresponding to it compared to the optical flow computed directly by the Horn and Schunck algorithm. We provide also a comparison to the optical flow ground truth using three kinds of error: average angular error (aae), standard angular error (stae), and endpoint error (epe). This is a good indirect way to evaluate scene flow computation results because the behavior of the Horn and Schunck method is a generally well-understood benchmark.

The formulation parameters were determined empirically. Distances are measured in pixels; the fronto-parallel plane position Z_0 has been fixed to 6×10^4 pixels. The camera focal length f has been approximated to 600 pixels (Sekkati and Mitiche, 2007); the initial value of scene flow and depth at each point are, respectively, 0 and Z_0. Coefficients α and β are given in the caption of each figure. Regularized differentiation's coefficient λ is fixed to 1 in all the examples.

In general, all of the proof-of-concept examples we show in the following support the validity of the scheme and its implementation. In the examples discussed below, one can make the following observations/conclusions:

- In all the examples, the scene flow and induced optical flow are consistent with the actual motion of the objects.
- The color-coded depth display is in line with the structure of the objects (i.e., the relative depth of object surfaces).
- The obtained optical flow is in keeping with the output of the well tested/researched benchmark algorithm of Horn and Schunck. It is worth noting here that the velocities we obtained are less noisy than those computed with Horn and Schunck algorithm. This can be explained by the fact that our method benefited from the use of (i) 3D information and (ii) better estimates of the image spatiotemporal derivatives via regularized differentiation.
- In all examples, the corresponding anaglyphs offered viewers a strong sense of depth.

5.1. Synthetic Squares Sequence

This is a sequence of two consecutive images of two overlapping squares moving against a moving background, to evaluate quantitatively the computed scene flow. This evaluation is done via the image motion that it induces. This induced motion will be compared to the actual image motion and the motion computed by the Horn and Schunck algorithm. The actual image motions are, in pixels: (−1,−1) for the upper square, (1,1) for the lower, and (0,−1) for the background. Noise has been added independently in the first and second image. Noise values are from a discretized, shifted, truncated Gaussian in the interval between 0 and 100 gray levels, within the overall range 0–255 of the image. The first of the two images is shown in the leftmost display of **Figure 3**; the vector-coded ground truth and the computed image motion are displayed in the second and third images, respectively; the results with the Horn and Schunck method are shown in the rightmost image. In general, such vector displays are meant to

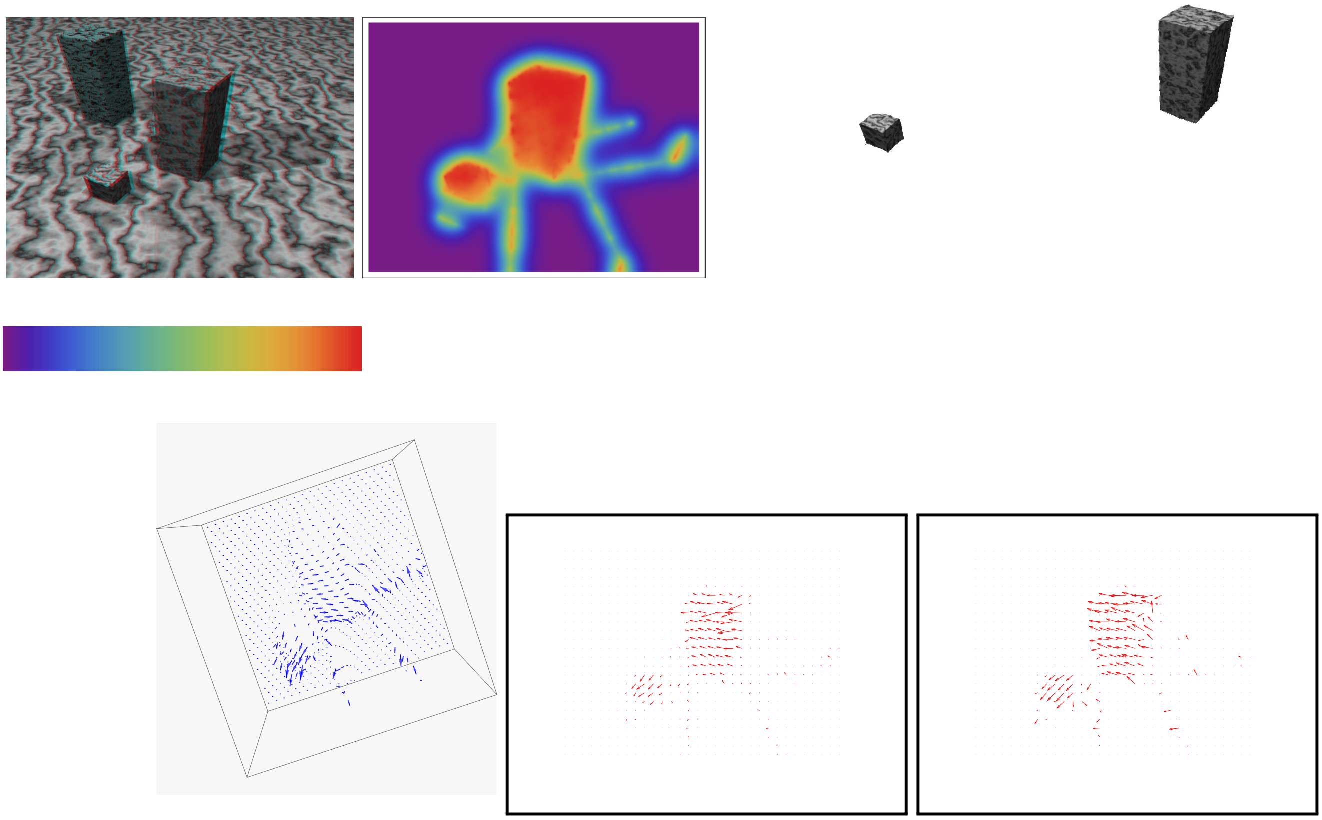

FIGURE 4 | *Marbled-blocks* sequence results (better perceived when figures are enlarged on screen). Parameters: $\alpha = 6 \times 10^7$ and $\beta = 10^2$. First row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth along with the used color palette, with depth increasing from right (red) to left (purple); novel viewpoint images of the two moving blocks. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed directly by the Horn and Schunck algorithm.

reassure that the image motion is visually consistent with its expected overall appearance. Quantitatively, the average angular error for the image motion induced by the computed scene flow is 15° and the average error on the length is 0.4 pixel. The Horn and Schunck algorithm was overwhelmed by the image noise; its average angular error is 42° and the average error on the length is 1 pixel. The proposed scheme has performed better than the Horn and Schunck algorithm because it used subsuming higher level 3D information, from which image motion can be recovered point-wise according to model Eq. 2, as well as a better estimate of the image spatiotemporal derivatives via regularized differentiation.

5.2. Marbled-Block Sequence

In this example, we use the *Marbled-block* synthetic sequence from the database of KOGS/IAKS Laboratory, Germany. There are three blocks in this sequence, two of which are moving. The rightmost block moves in depth to the left and the one in the middle moves forward to the left. There are aspects, which make 3D interpretation challenging: the blocks have a macro texture of weak spatiotemporal intensity variations within the textons and similar to the texture of the floor. As a result, the occluding boundaries of the blocks are ill defined at places. The blocks also cast shadows which move. Results are shown in **Figure 4**. The first row depicts (from left to right): an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth along with the used color palette[1]; and novel viewpoint images of the two moving blocks. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed directly by the Horn and Schunck algorithm.

5.3. Cylinder and Boxes Sequence

This second example uses a real image sequence [courtesy of Debrunner and Ahuja (1998)], shown in **Figure 5**. This sequence depicts three moving objects: a box moving to the right at an image rate of about 0.30 pixel per frame; a cylindrical surface rotating about a vertical axis at a velocity of one degree per frame, and moving laterally to the right at an image rate of about 0.15 pixel per frame and, finally, a flat background moving to the right (parallel to the box motion) at approximately 0.15 pixel per frame. In this example, the 3D interpretation and recovery is hard because of its unhelpful 3D motion. Results are displayed in **Figure 5**: first row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image

[1] We used the same color palette for all examples, with depth increasing from right (red) to left (purple)

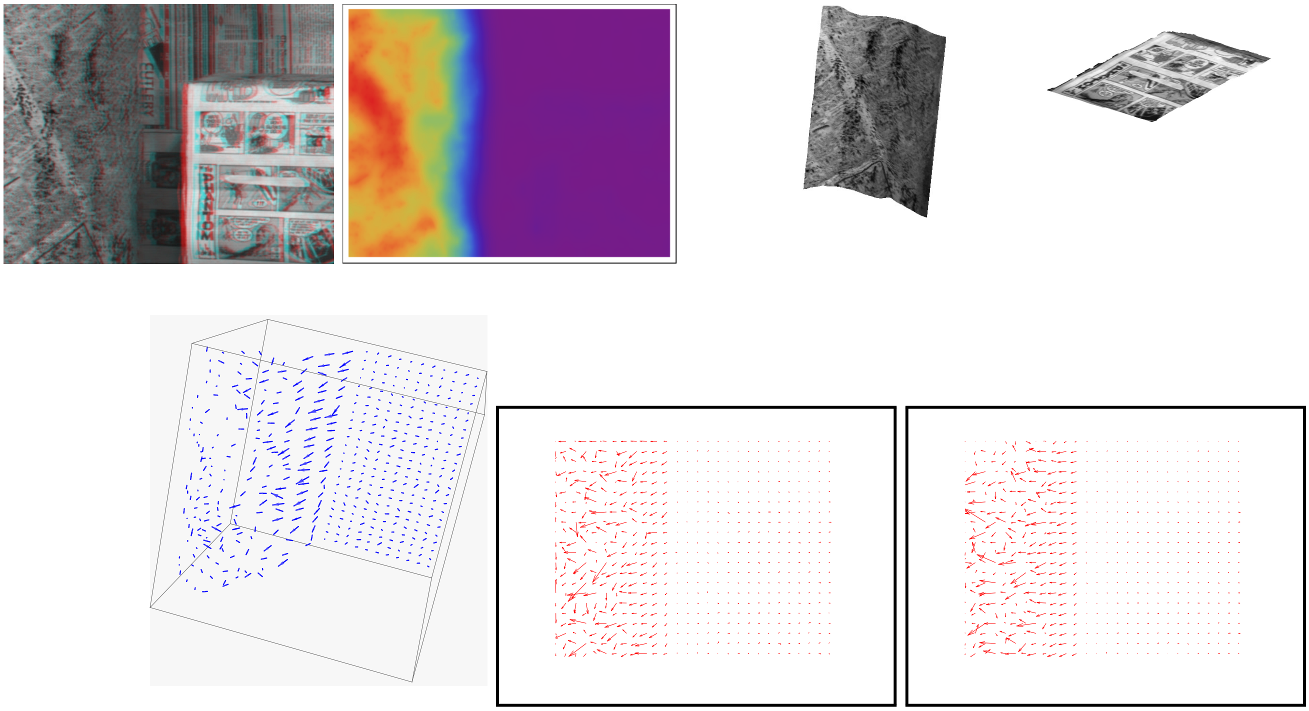

FIGURE 5 | *Cylinder and box* sequence results (better perceived when figures are enlarged on screen). Parameters: $\alpha = 6 \times 10^6$ and $\beta = 10^4$. First row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth; novel viewpoint images the cylindrical surface and the box. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed by the Horn and Schunck algorithm.

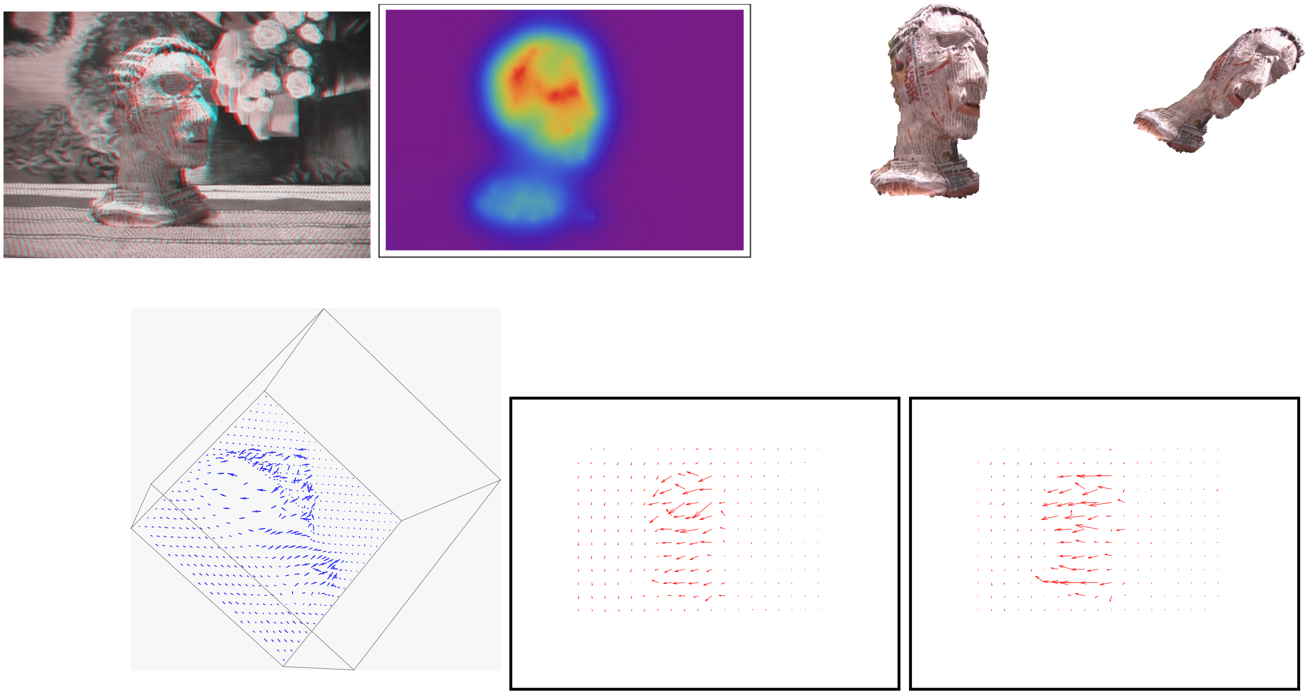

FIGURE 6 | *Berber* figurine sequence results (better perceived when figures are enlarged on the screen). Parameters: $\alpha = 6 \times 10^7$; $\beta = 5 \times 10^4$. First row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth; novel viewpoint images of the figurine. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed by the Horn and Schunck algorithm.

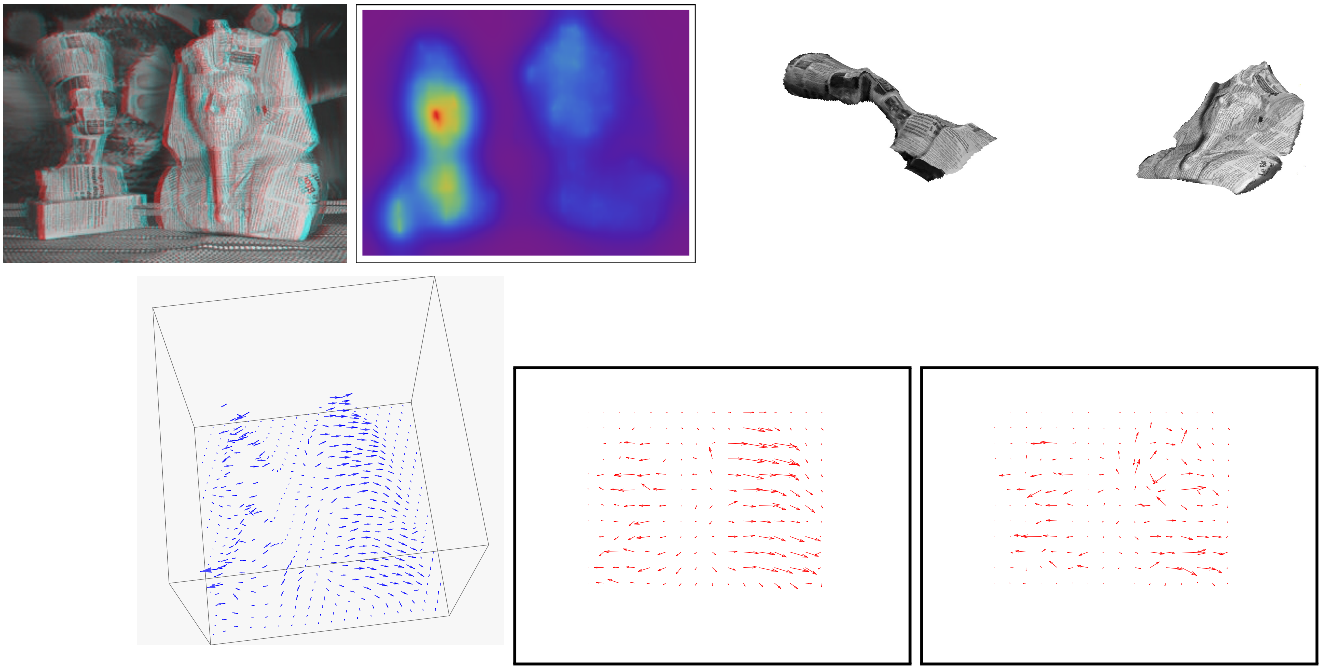

FIGURE 7 | *Pharaohs* figurines sequence (better perceived when figures are enlarged on screen). Parameters: $\alpha = 6 \times 10^7$; $\beta = \times 10^2$. First row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth; novel viewpoint images of the figurines. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow; optical flow computed by the Horn and Schunck algorithm.

sequence; a color-coded display of the recovered depth; novel viewpoint images of the cylindrical surface and the box. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed by the Horn and Schunck algorithm.

5.4. Berber Sequence

This example uses the *Berber* real sequence. The figurine rotates about a nearly vertical axis and moves forward to the left in a static environment. **Figure 6** displays the results: first row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth; novel viewpoint images of the figurine. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (2); the optical flow computed by the Horn and Schunck algorithm.

5.5. Pharaohs Sequence

This example uses the *Pharaohs* real image sequence. There are two moving figurines in a static environment; the leftmost translates left and forward; the rightmost rotates about a nearly vertical axis to the right. Results are shown in **Figure 7**: first row from left to right: an anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth; novel viewpoint images of the figurines. Second row: a view of the scene flow vectors; optical flow corresponding to the estimated scene flow; optical flow computed by the Horn and Schunck algorithm.

TABLE 1 | Average angular error (aae), standard angular error (stae), and endpoint error (epe) for the optical flow corresponding to the estimated scene flow using regularized differentiation (RD) vs. optical flow computed directly by the Horn and Schunck algorithm (HS).

	Errors	RD	HS
Marbled-block	aae	4.14	4.09
	stae	8.56	8.73
	epe	0.10	0.08
Cylinder	aae	18.68	15.72
	stae	18.91	18.42
	epe	0.78	0.68
Berber	aae	11.61	10.18
	stae	10.16	10.50
	epe	0.41	0.35
Pharaohs	aae	20.01	27.87
	stae	19.21	27.83
	epe	0.62	0.82

Coefficient λ was fixed equal to 1 for all the examples.

The results for scene flow are shown in **Table 1** for each of the examples described above.

6. Conclusion and Discussion

The goal of this study was concurrent recovery of scene flow and depth from a monocular image sequence. We developed a variational method, which minimizes a functional containing a data term of joint scene flow and depth conformity to the image sequence spatiotemporal variations, and quadratic smoothness regularization terms. The data term follows rewriting optical flow as a function of scene flow and depth in the classical optical flow gradient constraint of Horn and Schunck. As a result, the formulation is analogous to the classical Horn and Schunck optical flow estimation method, except that it involves the variables of scene flow and depth rather than image motion. Monocular processing is a unique feature of this scheme because previous scene flow recovery schemes have used binocular image sequences rather than a single image stream as in this study.

Another characteristic is the occurrence of both depth and scene flow as unknowns in the equations used to state the problem. The variational paradigm fitted naturally with these equations to give a single optimization formulation, free from the intervention of outside processes since all the relevant variables, namely depth and scene flow coordinates, occur simultaneously. As a result, the formulation translates into a tractable algorithm whose behavior can be explained. This algorithm follows the discretization of the objective functional Euler–Lagrange equations, giving a large scale sparse system of linear equations in the unknowns of depth and the three scene flow coordinates. The equations can be ordered in such a way that its matrix is symmetric positive definite such that they can be solved efficiently by Gauss–Seidel iterations.

The focus of this study being on the formulation proper, it was sufficient to use Gauss–Seidel iterations in the proof-of-concepts examples that we described in the experimental section. However, one can explore other schemes for more efficient numerical resolution as the literature on large sparse systems of linear equations is quite vast. For instance, one can investigate (Ciarlet, 1982) classical convergence acceleration of the Gauss–Seidel by successive over-relaxation, or an iterative scheme designed for positive definite systems, such as the conjugate gradient algorithm. The sequential subspace correction (SSC) method (Hackbush, 1994), which would process the minimization in the four independent linear subspaces of depth and scene flow coordinates sequentially can also prove to be quite efficient; for each subspace, the Gauss–Seidel iterations can be used. The SSC can be parallelized. There is also a rich literature on efficient modern Krylov subspace methods where a matrix need only be specified as a matrix–vector operator (Simoncini and Szyld, 2007).

This study used Tikhonov regularization for scene flow and depth. It did not affect the purpose of formulating monocular recovery of these variables. However, quadratic regularization smooths the variables recovered at and in the proximity of their discontinuities, namely sharp changes in depth and motion boundaries. There are several ways of specifying boundary preserving recovery (Mitiche and Aggarwal, 2013). For instance, one can use the Aubert et al. function, in place of the quadratic function, or simply the L^1 norm. In addition to preserving discontinuities, the L^1 norm can be approximated in practice for faster computation without affecting accuracy in a noticeable way. Both motion and depth discontinuities can also be preserved by concurrent motion computation and segmentation (Mitiche and Sekkati, 2006).

The examples we gave are for proof of concept only. They show that the formulation is sound, correctly implemented, but has obvious limitations, such as boundary blurring interpretation and use of approximate camera parameters. Nevertheless, the results clearly indicate that the method is worthy of further investigation. We are currently extending it to account for motion and depth discontinuities via L^1 regularization. We are also investigating joint motion segmentation and estimation, an extension to scene flow and depth of the scheme in (Mitiche and Sekkati, 2006). Experimental validation must be based on a larger database of three-dimensional moving objects of various geometries that would test various difficulties, such as motion and depth discontinuities, motion of large extent, and image noise and resolution in common practical settings. In particular, quantitative validation will require computer graphics generation of appropriate synthetic objects in motion for which ground truth scene flow can be calculated.

References

Adiv, G. (1985). Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 7, 384–401. doi:10.1109/TPAMI.1985.4767678

Aubert, G., Deriche, R., and Kornprobst, P. (1999). Computing optical flow via variational techniques. *SIAM J. Appl. Math.* 60, 156–182. doi:10.1137/S0036139998340170

Basha, T., Moses, Y., and Kiryati, N. (2013). Multi-view scene flow estimation: a view centered variational approach. *Int. J. Comput. Vis.* 101, 6–21. doi:10.1007/s11263-012-0542-7

Brodsky, T., Fermuller, C., and Aloimonos, Y. (2000). Structure from motion: beyond the epipolar constraint. *Int. J. Comput. Vis.* 37, 231–258. doi:10.1023/A:1008132107950

Bruss, A. R., and Horn, B. K. P. (1983). Passive navigation. *Comput. Vis. Graph. Image Process.* 21, 3–20. doi:10.1016/S0734-189X(83)80026-7

Chartrand, R. (2005). *Numerical Differentiation of Noisy, Nonsmooth Data*. Los Alamos National Laboratory.

Ciarlet, P. (1982). "Introduction à l'analyse numérique matricielle et à l'optimisation," in *Collection Mathématiques appliquées pour la maîtrise* (Masson).

Cullum, J. (1971). Numerical differentiation and regularization. *SIAM J. Numer. Anal.* 8, 254–265. doi:10.1137/0708026

De Micheli, E., and Giachero, F. (1994). "Motion and structure from one dimensional optical flow," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* (Seatle, WA: IEEE), 962–965.

Debrunner, C., and Ahuja, N. (1998). Segmentation and factorization-based motion and structure estimation for long image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 206–211. doi:10.1109/34.659941

Deriche, R., Kornprobst, P., and Aubert, G. (1995). "Optical-flow estimation while preserving its discontinuities: a variational approach," in *ACCV, Volume 1035 of Lecture Notes in Computer Science*, eds S. Z. Li, D. P. Mital, E. K. Teoh, and H. Wang (Singapore: Springer), 71–80.

Fejes, S., and Davis, L. S. (1998). "What can projections of flow fields tell us about visual motion," in *ICCV* (Bombay: IEEE), 979–986.

Forsythe, G. E., Malcolm, M. A., and Moler, C. B. (1977). "Computer methods for mathematical computations," in *Prentice-Hall Series in Automatic Computation* (Englewood Cliffs, NJ: Prentice-Hall).

Gupta, N., and Kanal, N. (1995). 3-D motion estimation from motion field. *Artif. Intell.* 78, 45–86. doi:10.1016/0004-3702(95)00031-3

Hackbush, W. (1994). *Iterative Solution of Large Sparse Systems of Equations*, Vol. 95. Springer.

Hanke, M., and Scherzer, O. (2001). Inverse problems light: numerical differentiation. *Am. Math. Mon.* 108, 512–521. doi:10.2307/2695705

Heeger, D. J., and Jepson, A. D. (1992). Subspace methods for recovering rigid motion I: algorithm and implementation. *Int. J. Comput. Vis.* 7, 95–117. doi:10.1007/BF00128130

Hildreth, E. C. (1984). *The Measurement of Visual Motion*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA; London.

Horn, B. K. P., and Schunck, B. G. (1981). Determining optical flow. *Artif. Intell.* 17, 185–203. doi:10.1016/0004-3702(81)90024-2

Horn, B. K. P., and Weldon, E. J. (1988). Direct methods for recovering motion. *Int. J. Comput. Vis.* 2, 51–76. doi:10.1007/BF00836281

Huguet, F., and Devernay, F. (2007). "A variational method for scene flow estimation from stereo sequences," in *IEEE International Conference on Computer Vision (ICCV)* (Rio de Janeiro: IEEE), 1–7.

Hung, Y. S., and Ho, H. T. (1999). A Kalman filter approach to direct depth estimation incorporating surface structure. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 570–575. doi:10.1109/34.771330

Liu, H., Chellappa, R., and Rosenfeld, A. (2002). "A hierarchical approach for obtaining structure from two-frame optical flow," in *Proceedings of the Workshop on Motion and Video Computing, MOTION '02* (Washington, DC: IEEE Computer Society), 214–219.

Longuet-Higgins, H. C., and Prazdny, K. (1980). The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B Biol. Sci.* 208, 385–397. doi:10.1098/rspb.1980.0057

MacLean, W. J., Jepson, A. D., and Frecker, R. C. (1994). "Recovery of egomotion and segmentation of independent object motion using the em algorithm," in *British Machine Vision Conference, BMVC*, ed. E. R. Hancock (New York, UK: BMVA Press), 1–10.

Mitiche, A., and Aggarwal, J. (2013). *Computer Vision Analysis of Image Motion by Variational Methods*. Springer.

Mitiche, A., and Hadjres, S. (2003). Mdl estimation of a dense map of relative depth and 3D motion from a temporal sequence of images. *Pattern Anal. Appl.* 6, 78–87. doi:10.1007/s10044-002-0182-6

Mitiche, A., and Sekkati, H. (2006). Optical flow 3D segmentation and interpretation: a variational method with active curve evolution and level sets. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 1818–1829. doi:10.1109/TPAMI.2006.232

Pons, J.-P., Keriven, R., Faugeras, O., and Hermosillo, G. (2003). "Variational stereovision and 3d scene flow estimation with statistical similarity measures," in *IEEE International Conference On Computer Vision (ICCV)* (Nice: IEEE), 597–602.

Rabe, C., Müller, T., Wedel, A., and Franke, U. (2010). "Dense, robust, and accurate motion field estimation from stereo image sequences in real-time," in *Proceedings of the 11th European Conference on Computer Vision, Volume 6314* of *Lecture Notes in Computer Science*, eds K. Daniilidis, P. Maragos, and N. Paragios (Heraklion: Springer), 582–595.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik* 10, 177–183. doi:10.1007/BF02162161

Schoenberg, I. J. (1964). Spline functions and the problem of graduation. *Proc. Natl. Acad. Sci.U.S.A.* 54, 947–950. doi:10.1073/pnas.52.4.947

Sekkati, H., and Mitiche, A. (2006a). Concurrent 3-D motion segmentation and 3-D interpretation of temporal sequences of monocular images. *IEEE Trans. Image Process.* 15, 641–653. doi:10.1109/TIP.2005.863699

Sekkati, H., and Mitiche, A. (2006b). Joint optical flow estimation, segmentation, and 3D interpretation with level sets. *Comput. Vis. Image Underst.* 103, 89–100. doi:10.1016/j.cviu.2005.11.002

Sekkati, H., and Mitiche, A. (2007). A variational method for the recovery of dense 3D structure from motion. *Rob. Auton. Syst.* 55, 597–607. doi:10.1016/j.robot.2006.11.006

Shahraray, B., and Brown, M. (1988). "Robust depth estimation from optical flow," in *International Conference on Computer Vision, ICCV* (Tampa, FL: IEEE), 641–650.

Simoncini, V., and Szyld, D. B. (2007). Recent computational developments in Krylov subspace methods for linear systems. *Numer. Lin. Algebra Appl.* 14, 1–59. doi:10.1002/nla.499

Srinivasan, S. (2000). Extracting structure from optical flow using the fast error search technique. *Int. J. Comput. Vis.* 37, 203–230. doi:10.1023/A:1008111923880

Stoer, J., and Bulirsch, R. (2002). *Introduction to Numerical Analysis. Texts in Applied Mathematics; 12*, 3rd Edn. New York, NY: Springer.

Taalebinezhaad, M. A. (1992). Direct recovery of motion and shape in the general case by fixation. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 847–853. doi:10.1109/34.149584

Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 413–424. doi:10.1109/TPAMI.1986.4767807

Ullman, S. (1983). *Computational Studies in the Interpretation of Structure and Motion: Summary and Extension*. Technical Report, MIT.

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (2005). Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 475–480. doi:10.1109/TPAMI.2005.63

Vogel, C., Schindler, K., and Roth, S. (2013). "Piecewise rigid scene flow," in *IEEE International Conference on Computer Vision, ICCV 2013* (Sydney, NSW: IEEE), 1377–1384.

Vogel, C. R. (2002). *Computational Methods for Inverse Problems*. SIAM Frontiers in Applied Mathematics.

Weber, J., and Malik, J. (1997). Rigid body segmentation and shape description from dense optical flow under weak perspective. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 139–143. doi:10.1109/34.574794

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., and Cremers, D. (2011). Stereoscopic scene flow computation for 3D motion understanding. *Int. J. Comput. Vis.* 95, 29–51. doi:10.1007/s11263-010-0404-0

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., and Cremers, D. (2008). "Efficient dense scene flow from sparse or dense stereo data," in *European Conference on Computer Vision (ECCV)*, Vol. 1 (Marseille: Springer), 739–751.

Xiong, Y., and Shafer, S. (1995). *Dense Structure from a Dense Optical Flow*. Technical Report CMU-RI-TR-95-10. Pittsburgh, PA: Robotics Institute.

Zhang, Y., and Kambhamettu, C. (2000). "Integrated 3D scene flow and structure recovery from multiview image sequences," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2 (Hilton Head, SC: IEEE), 674–681.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.