



Monocular, Boundary-Preserving Joint Recovery of Scene Flow and Depth

Yosra Mathlouthi¹, Amar Mitiche^{1*} and Ismail Ben Ayed²

¹ Institut National de la Recherche Scientifique (INRS-EMT), Montreal, QC, Canada, ² Ecole de Technologie Supérieure (ETS), Montreal, QC, Canada

Variational joint recovery of scene flow and depth from a single image sequence, rather than from a stereo sequence as others required, was investigated in Mitiche et al. (2015) using an integral functional with a term of conformity of scene flow and depth to the image sequence spatiotemporal variations, and L^2 regularization terms for smooth depth field and scene flow. The resulting scheme was analogous to the Horn and Schunck optical flow estimation method, except that the unknowns were depth and scene flow rather than optical flow. Several examples were given to show the basic potency of the method: it was able to recover good depth and motion, except at their boundaries because L^2 regularization is blind to discontinuities which it smooths indiscriminately. The method that we study in this paper generalizes to L^1 regularization the formulation of Mitiche et al. (2015) so that it computes boundary-preserving estimates of both depth and scene flow. The image derivatives, which appear as data in the functional, are computed from the recorded image sequence also by a variational method, which uses L^1 regularization to preserve their discontinuities. Although L^1 regularization yields non-linear Euler–Lagrange equations for the minimization of the objective functional, these can be solved efficiently. The advantages of the generalization, namely, sharper computed depth and three-dimensional motion, are put in evidence in experimentation with real and synthetic images, which shows the results of L^1 versus L^2 regularization of depth and motion, as well as the results using L^1 rather than L^2 regularization of image derivatives.

Keywords: scene flow, image sequence analysis, 3D motion, depth, L^1 regularization, image derivatives

OPEN ACCESS

Edited by:

Kaleem Siddiqi,
McGill University, Canada

Reviewed by:

K. C. Santosh,
University of South Dakota, USA
Scott P. McCloskey,
Honeywell, USA

*Correspondence:

Amar Mitiche
mitiche@emt.inrs.ca

Specialty section:

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

Received: 06 April 2016

Accepted: 15 September 2016

Published: 30 September 2016

Citation:

Mathlouthi Y, Mitiche A and
Ben Ayed I (2016) Monocular,
Boundary-Preserving Joint Recovery
of Scene Flow and Depth.
Front. ICT 3:21.
doi: 10.3389/fict.2016.00021

1. INTRODUCTION

Scene flow is the three-dimensional (3D) motion field of the visible environmental surfaces projected on the image domain: at each image point, scene flow is the 3D velocity of the corresponding environmental surface point. It is the time derivative of 3D position. As such, it is a function of both depth and optical flow (Longuet-Higgins and Prazdny, 1981). Scene flow computation has been the focus of several recent studies (Vedula et al., 2005; Hugué and Devernay, 2007; Pons et al., 2007; Wedel et al., 2008; Mitiche and Aggarwal, 2013; Vogel et al., 2013). It is a typical inverse problem best stated by variational formulations (Mitiche and Aggarwal, 2013). Basic variational statements use L^2 (Tikhonov) regularization. This regularization, which imposes smoothness on the solution, yields linear terms in the Euler–Lagrange equations. This is the case with the scheme described in

Mitiche et al. (2015) and of which we describe a generalization in this paper. The scheme improved significantly on others because it needed a single image sequence rather than a stereo stream like other methods. Also, it formulated the problem using an integral functional of two terms: a data fidelity term to constrain the computed scene flow to conform to the image sequence spatiotemporal derivatives and L^2 regularization terms to constrain the computed scene flow and depth to be smooth, which led to linear Euler–Lagrange equations for the minimization of the objective functional, much like in the Horn and Schunck optical flow formulation (Horn and Schunck, 1981), except that it involved depth and scene flow rather than optical flow. However, L^2 regularization blurs the computed scene flow and depth boundaries because it imposes their smoothness everywhere. In general, some form of boundary-preserving regularization is necessary. This is particularly true when environmental objects move independently relative to the viewing system. The variation of motion and structure in such a case can be sharp and significant at these objects occluding boundaries, and therefore, accuracy would require that they be preserved by the regularization operator.

Boundary-preserving recovery of scene flow and depth can be specified in various ways (Mitiche and Aggarwal, 2013). For example, instead of the L^2 norm, one can use the Aubert et al. function (Aubert et al., 1999; Aubert and Kornprobst, 2002) or the L^1 norm. We can also inhibit smoothing across boundaries by joint 3D motion estimation and segmentation (Mitiche and Sekkati, 2006). In this follow-up study of our previous investigation (Mitiche et al., 2015), we apply the L^1 regularization. There are three basic reasons for this choice: (1) the ability of the L^1 constraint to preserve sharp boundaries while penalizing oscillations. This is well suited for “blocky images” which tend to be smooth inside regions, which have sharp significant boundaries, as is, in general, typical of motion fields; (2) in practice, it can be implemented by computationally efficient approximations without affecting the accuracy of results in a noticeable way; and (3) there is a significant body of literature in its support, particularly in image restoration (Vogel, 2002).

Recovery of scene flow and depth uses the image sequence spatiotemporal derivatives. The study in Mitiche et al. (2015) computed derivatives by a variational formulation, which used an anti-differentiation data fidelity term, and L^2 regularization. The anti-differentiation term expressed the fact that an image derivative is a function which when integrated gives the image. The present study uses the same anti-differentiation data fidelity term but substitutes L^1 regularization for L^2 in order, first, to be consistent with the use of L^1 regularization of scene flow and depth and, second, to have the evaluation of the derivatives account for their discontinuities.

We conducted several experiments with real and synthetic data to verify the validity and efficiency of the scheme. We show comparative results that put in evidence the improvements one can obtain by using L^1 rather than L^2 regularization of depth and motion, as well as by evaluating the image derivatives with L^1 rather than L^2 regularization.

The remainder of this paper is organized as follows: Section 2 develops the objective functional, and Section 3 describes its

minimization. Section 4 shows how the image derivatives are computed. The validation experiments, using synthetic and real image sequences, are described in Section 5. Section 6 contains a conclusion.

2. FORMULATION

Let $I: (x,y,t) \rightarrow I(x,y,t)$ be an image sequence, where (x,y) are the spatial coordinates on the bounded image domain Ω , and $t \in \mathbf{R}^+$ designates time. Let (X,Y,Z) be the coordinates of a point \mathbf{P} in space and (x,y) the coordinates of its projection. The coordinate system and the imaging geometry are shown in **Figure 1**. Let U , V , and W be the scene flow coordinate functions. The scene flow and depth linear gradient constraint (Mitiche et al., 2015), which relates the scene flow coordinates U , V , and W and depth Z to the spatiotemporal image is

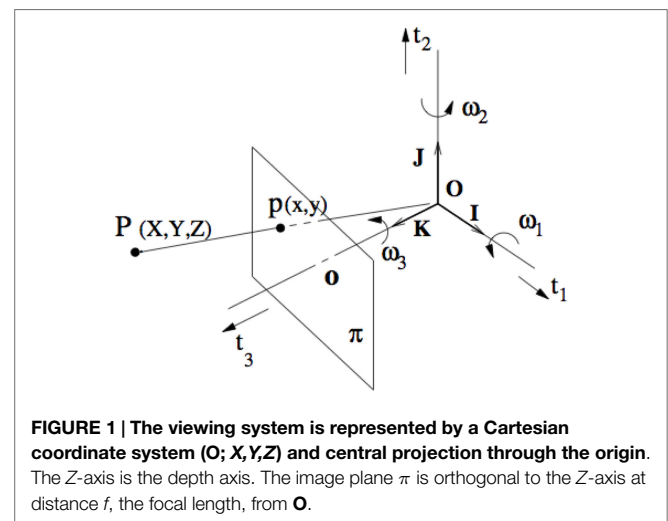
$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ = 0, \quad (1)$$

where I_x , I_y , and I_t are the image spatiotemporal derivatives, Z designates depth (**Figure 1**), and $(U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ is the scene flow vector at \mathbf{P} . This is a homogeneous linear equation in the variables of scene flow and depth. The homogeneity results from the aperture problem. Multiplication of motion and depth (and structure thereof) by a constant (scale) maintains the equation integrity. One can remove this uncertainty of scale by choosing the depth to be *relative* to the frontoparallel plane $Z = Z_0$, for some positive depth Z_0 (Mitiche et al., 2015). Therefore, equation (1) becomes:

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_t(Z - Z_0) + I_tZ_0 = 0 \quad (2)$$

For notational convenience and economy, we will reuse the symbol Z for depth *relative* to the frontoparallel plane $Z = Z_0$, in which case we can write equation (2) as

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0 \quad (3)$$



In the formulation of Mitiche et al. (2015), scene flow and relative depth resulted from the minimization of the following L^2 smoothness regularization functional:

$$\begin{aligned} \mathbf{E}(U, V, W, Z|I) = & \frac{1}{2} \int_{\Omega} (f_{I_x}U + f_{I_y}V - (xI_x + yI_y)W \\ & + I_tZ + I_tZ_0)^2 dx dy \\ & + \frac{\alpha}{2} \int_{\Omega} (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2) dx dy \\ & + \frac{\beta}{2} \int_{\Omega} \|\nabla Z\|^2 dx dy, \end{aligned} \quad (4)$$

where α and β scale the relative contribution of the terms of the functional. For boundary preservation, we replace the L^2 norm regularization term of each variable by an L^1 norm term, namely: $\int_{\Omega} \|\nabla Q\| dx dy = \int_{\Omega} (Q_x^2 + Q_y^2)^{\frac{1}{2}} dx dy$, where $Q \in \{U, V, W, Z\}$. Therefore, the objective functional is

$$\begin{aligned} \mathbf{E}(U, V, W, Z|I) = & \frac{1}{2} \int_{\Omega} (f_{I_x}U + f_{I_y}V - (xI_x + yI_y)W \\ & + I_tZ + I_tZ_0)^2 dx dy \\ & + \frac{\alpha}{2} \int_{\Omega} ((U_x^2 + U_y^2)^{\frac{1}{2}} + (V_x^2 + V_y^2)^{\frac{1}{2}} \\ & + (W_x^2 + W_y^2)^{\frac{1}{2}}) dx dy \\ & + \frac{\beta}{2} \int_{\Omega} (Z_x^2 + Z_y^2)^{\frac{1}{2}} dx dy, \end{aligned} \quad (5)$$

3. OPTIMIZATION

The Euler–Lagrange equations corresponding to functional equation (5) are

$$\begin{aligned} & f_{I_x}(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \alpha \frac{\partial}{\partial x} \frac{U_x}{(U_x^2 + U_y^2)^{\frac{1}{2}}} - \alpha \frac{\partial}{\partial y} \frac{U_y}{(U_x^2 + U_y^2)^{\frac{1}{2}}} = 0 \\ & f_{I_y}(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \alpha \frac{\partial}{\partial x} \frac{V_x}{(V_x^2 + V_y^2)^{\frac{1}{2}}} - \alpha \frac{\partial}{\partial y} \frac{V_y}{(V_x^2 + V_y^2)^{\frac{1}{2}}} = 0 \\ & (xI_x + yI_y)(f_{I_x}U + f_{I_y}V - (xI_x + yI_y)W + I_tZ + I_tZ_0) \\ & + \alpha \frac{\partial}{\partial x} \frac{W_x}{(W_x^2 + W_y^2)^{\frac{1}{2}}} - \alpha \frac{\partial}{\partial y} \frac{W_y}{(W_x^2 + W_y^2)^{\frac{1}{2}}} = 0 \\ & I_t(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \beta \frac{\partial}{\partial x} \frac{Z_x}{(Z_x^2 + Z_y^2)^{\frac{1}{2}}} - \beta \frac{\partial}{\partial y} \frac{Z_y}{(Z_x^2 + Z_y^2)^{\frac{1}{2}}} = 0, \end{aligned} \quad (6)$$

with the Neumann boundary conditions:

$$\frac{\partial U}{\partial \mathbf{n}} = 0, \quad \frac{\partial V}{\partial \mathbf{n}} = 0, \quad \frac{\partial W}{\partial \mathbf{n}} = 0, \quad \frac{\partial Z}{\partial \mathbf{n}} = 0, \quad (7)$$

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

Were it not for the denominator of the regularization terms in equation (6), the equations would be linear. It is common in numerical analysis to solve such systems of equations using an iterative method where the non-linear terms are evaluated at the preceding iteration, in which case they are treated as data at the current iteration, and the equations to solve are linear. In our case, we alternate the following two steps at the current k -th iteration.

Step 1: this step accounts of the non-linearity of the obtained Euler–Lagrange equations and consists of updating the denominators of the regularization terms in equation (6) at each grid point:

$$g(Q^k) = \frac{1}{((Q_x^{k-1})^2 + (Q_y^{k-1})^2 + \varepsilon)^{\frac{1}{2}}}, \quad Q \in \{U, V, W, Z\} \quad (8)$$

where ε is a small positive value whose purpose is to remedy the non-differentiability of the Euclidean norm at the origin, without affecting the computational outcome in a noticeable way. Notice that the pointwise updates in equation (8) have a computational complexity that grows *linearly* with respect to N (the image size): the complexity is N multiplied by the fixed complexity of 4 updates of the form in equation (8).

Step 2: with pointwise coefficients $g(Q^k)(Q \in \{U, V, W, Z\})$ fixed, this step is an update (iteration k) of variables $\{U, V, W, Z\}$ for solving the following linear system:

$$\begin{aligned} & f_{I_x}(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \alpha g(U^k) \nabla^2 U = 0 \\ & f_{I_y}(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \alpha g(V^k) \nabla^2 V = 0 \\ & (xI_x + yI_y)(f_{I_x}U + f_{I_y}V - (xI_x + yI_y)W + I_tZ + I_tZ_0) \\ & - \alpha g(W^k) \nabla^2 W = 0 \\ & I_t(f_{I_x}U + f_{I_y}V + (-xI_x - yI_y)W + I_tZ + I_tZ_0) \\ & - \beta g(Z^k) \nabla^2 Z = 0, \end{aligned} \quad (9)$$

The four equations of equation (9) are written for each point of image domain Ω . Let Ω be discretized via a unit-spacing grid, and let the grid points be indexed by the integers $\{1, 2, \dots, N\}$. The pixel numbering is according to the lexicographical order, i.e., by scanning the image top-down and left to right. If the image is of size $n \times n$, then $N = n^2$. Let $a = f_{I_x}$, $b = f_{I_y}$, $c = -(xI_x + yI_y)$, and $d = I_t$.

For all grid point indices $i \in \{1, 2, \dots, N\}$, a discrete approximation of the linear system (equation (9)) is

$$(S) \begin{cases} (a_i^2 + \alpha g_i(U^k)n_i)U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i \\ - \alpha g_i(U^k) \sum_{j \in \mathbf{N}_i} U_j = -a_i d_i Z_0 \\ b_i a_i U_i + (b_i^2 + \alpha g_i(V^k)n_i)V_i + b_i c_i W_i + b_i d_i Z_i \\ - \alpha g_i(V^k) \sum_{j \in \mathbf{N}_i} V_j = -b_i d_i Z_0 \\ c_i a_i U_i + c_i b_i V_i + (c_i^2 - \alpha g_i(W^k)n_i)W_i + c_i d_i Z_i \\ - \alpha g_i(W^k) \sum_{j \in \mathbf{N}_i} W_j = -c_i d_i Z_0 \\ d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + (d_i^2 + \beta g_i(Z^k)n_i)Z_i \\ - \beta g_i(Z^k) \sum_{j \in \mathbf{N}_i} Z_j = -d_i^2 Z_0, \end{cases}$$

where $(U_i, V_i, W_i, Z_i) = (U, V, W, Z)_i$ is the scene flow vector at grid point i ; a_i, b_i, c_i , and d_i are the values at i of a, b, c , and d , respectively; $g_i(Q^k)$, $Q \in \{U, V, W, Z\}$, are the pointwise updates of equation (8) evaluated at i , and \mathbf{N}_i is the set of indices of the neighbors of i . For the 4-neighborhood, $n_i = \text{card}(\mathbf{N}_i) = 4$ for points interior to the discrete image domain, and $n_i < 4$ for boundary image points. Laplacian $\nabla^2 Q$ in the Euler-Lagrange equations ($Q \in \{U, V, W, Z\}$) has been discretized as $\frac{1}{4} \sum_{j \in \mathbf{N}_i} (Q_j - Q_i)$, with α (respectively β) absorbing the factor $1/4$.

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

$$\mathbf{A}\mathbf{q} = \mathbf{r} \quad (10)$$

where \mathbf{A} is the $4N \times 4N$ matrix with elements $\mathbf{A}_{4i-3,4i-3} = a_i^2 + \alpha g_i(U^k)n_i$; $\mathbf{A}_{4i-2,4i-2} = b_i^2 + \alpha g_i(V^k)n_i$; $\mathbf{A}_{4i-1,4i-1} = c_i^2 + \alpha g_i(W^k)n_i$; $\mathbf{A}_{4i,4i} = d_i^2 + \beta g_i(Z^k)n_i$; $\mathbf{A}_{4i-3,4i-2} = \mathbf{A}_{4i-2,4i-3} = a_i b_i$; $\mathbf{A}_{4i-3,4i-1} = \mathbf{A}_{4i-1,4i-3} = a_i c_i$; $\mathbf{A}_{4i-3,4i} = \mathbf{A}_{4i,4i-3} = a_i d_i$; $\mathbf{A}_{4i-2,4i-1} = \mathbf{A}_{4i-1,4i-2} = b_i c_i$; $\mathbf{A}_{4i-2,4i} = \mathbf{A}_{4i,4i-2} = b_i d_i$; $\mathbf{A}_{4i-1,4i} = \mathbf{A}_{4i,4i-1} = c_i d_i$; for all $i \in \{1, \dots, N\}$, and $\mathbf{A}_{4i-3,4j-3} = -\alpha g_i(U^k)$; $\mathbf{A}_{4i-2,4j-2} = -\alpha g_i(V^k)$; $\mathbf{A}_{4i-1,4j-1} = -\alpha g_i(W^k)$; and $\mathbf{A}_{4i,4j} = -\beta g_i(Z^k)$, for all $i, j \in \{1, \dots, N\}$ such that $j \in \mathbf{N}_i$, all other elements being equal to zero. This is a large-scale sparse system of linear equations, which can be solved by iterative updates for sparse matrices (Ciarlet, 1994; Stoer and Bulirsch, 2002). It is easy to prove that symmetric matrix \mathbf{A} is positive definite (PD). This means that a fast solution of equation (10) can be obtained by convergent 4×4 block-wise Gauss-Seidel updates. To show that \mathbf{A} is PD, it suffices to perform some algebraic manipulations to write $\mathbf{q}^t \mathbf{A} \mathbf{q}$ for all $\mathbf{q} \in \mathbf{R}^{4N}, \mathbf{q} \neq 0$, as follows:

$$\begin{aligned} \mathbf{q}^t \mathbf{A} \mathbf{q} &= \sum_{i=1}^N (a_i U_i + b_i V_i + c_i W_i + d_i Z_i)^2 \\ &+ \alpha \sum_{i=1}^N \sum_{j \in \mathbf{N}_i; j > i} g_i(U^k) (U_i - U_j)^2 + g_i(V^k) (V_i - V_j)^2 \\ &+ g_i(W^k) (W_i - W_j)^2 \\ &+ \beta \sum_{i=1}^N \sum_{j \in \mathbf{N}_i; j > i} g_i(Z^k) (Z_i - Z_j)^2 > 0 \end{aligned} \quad (11)$$

The positive definiteness of \mathbf{A} implies that iterative pointwise and block-wise Gauss-Seidel and relaxation updates for system (equation (10)) converge (Ciarlet, 1994; Stoer and Bulirsch, 2002). For a 4×4 block division of \mathbf{A} , the Gauss-Seidel update (iteration k) for each grid point $i \in \{1, \dots, N\}$ is

$$\begin{aligned} &(a_i^2 + \alpha g_i(U^k)n_i)U_i^{k+1} + a_i b_i V_i^{k+1} + a_i c_i W_i^{k+1} + a_i d_i Z_i^{k+1} \\ &= -a_i d_i Z_0 + \alpha g_i(U^k) \left(\sum_{j \in \mathbf{N}_i; j < i} U_j^{k+1} + \sum_{j \in \mathbf{N}_i; j > i} U_j^k \right) \end{aligned}$$

$$\begin{aligned} &b_i a_i U_i^{k+1} + (b_i^2 + \alpha g_i(V^k)n_i)V_i^{k+1} + b_i c_i W_i^{k+1} + b_i d_i Z_i^{k+1} \\ &= -b_i d_i Z_0 + \alpha g_i(V^k) \left(\sum_{j \in \mathbf{N}_i; j < i} V_j^{k+1} + \sum_{j \in \mathbf{N}_i; j > i} V_j^k \right) \\ &c_i a_i U_i^{k+1} + c_i b_i V_i^{k+1} + (c_i^2 + \alpha g_i(W^k)n_i)W_i^{k+1} + c_i d_i Z_i^{k+1} \\ &= -c_i d_i Z_0 + \alpha g_i(W^k) \left(\sum_{j \in \mathbf{N}_i; j < i} W_j^{k+1} + \sum_{j \in \mathbf{N}_i; j > i} W_j^k \right) \\ &d_i a_i U_i^{k+1} + d_i b_i V_i^{k+1} + d_i c_i W_i^{k+1} + (d_i^2 + \beta g_i(Z^k)n_i)Z_i^{k+1} \\ &= -d_i^2 Z_0 + \beta g_i(Z^k) \left(\sum_{j \in \mathbf{N}_i; j < i} Z_j^{k+1} + \sum_{j \in \mathbf{N}_i; j > i} Z_j^k \right), \end{aligned} \quad (12)$$

For each point $I \in \{1, \dots, N\}$, we solve a 4×4 linear system of equations, which can be done efficiently by a singular value decomposition (SVD) (Forsythe et al., 1977). The computational complexity of this step grows *linearly* with respect to N : the complexity is N multiplied by the fixed complexity of a 4×4 SVD.

3.1. Computational Load: L^1 vs. L^2

Although L^1 regularization yields non-linear Euler-Lagrange equations for the minimization of our objective functional, these can be solved efficiently via the two-step scheme that we proposed above. The pointwise updates in equation (8) account for the non-linearity of the L^1 model (Step 1) and are an additional computational load in comparison to L^2 regularization. However, these updates have a complexity that grows *linearly* with respect to N (image size). Therefore, they do not increase the computational time substantially; see the computational times in **Table 1**. Step 2 has a computational complexity that is similar to the L^2 regularization of Mitiche et al. (2015): in both cases, we have block-wise (4×4) and relaxation updates for a large-scale sparse system of linear equations, with a symmetric positive definite matrix \mathbf{A} . The complexity of each iteration is N multiplied by the fixed complexity of a 4×4 singular value decomposition, and the positive definiteness of \mathbf{A} implies that the block-wise updates converge.

Table 1 reports the computation (CPU) times for both L^1 and L^2 regularizations, in the case of four different test images of different sizes. Columns 2 and 3 contain the overall CPU times. The third column gives the CPU time for the block-wise SVD

TABLE 1 | Computation times per iteration (in seconds).

Sequences (dimensions in pixels)	L^2 (overall)	L^1 (overall)	Step 2 (L^2/L^1)	Step 1 (L^1)
Berber (240 × 360)	1.86	2.21	1.86	0.35
Pharaohs (240 × 320)	1.65	1.96	1.65	0.31
Cylinder (384 × 500)	4.13	4.90	4.13	0.77
Marbled-block (384 × 512)	4.23	5.02	4.23	0.79

updates solving a large-scale sparse system. These SVD computations occur in both L^2 and L^1 formulations (Step 2). The last column reports the CPU times for the pointwise updates in equation (8). These updates appear only in the L^1 formulation (Step 1). The simulations were run on an Intel i7-4500U Processor (4M Cache, up to 3.00 GHz), using MATLAB (2014b version).

4. PARTIAL DERIVATIVES

In computer vision, image derivatives are often approximated by locally averaged finite differences to lessen the impact of noise (Horn and Schunk, 1981; Marshall, 2005; Periaswamy and Farid, 2006; Sun et al., 2010; Heinrich et al., 2012; Mitiche et al., 2015). However, such fixed-support low-pass filtering does not generally fit the noise profile and can, therefore, be ineffective. A more effective way is to state differentiation as a spatially regularized variational problem, as was done in Mitiche et al. (2015). The process advocated in Mitiche et al. (2015) looked at the derivative of an image as a function which, when integrated, gives the image function. Consequently, the objective functional contained an anti-differentiation data term, which evaluates the conformity of a derivative to the image by constraining the integration of this derivative to produce the image. The functional also contained an L^2 regularization term. In this paper, we investigate a generalization, which accounts for derivative discontinuities using an L^1 regularization term because L^2 regularization constrains the image derivatives to be smooth everywhere and, as a result, would adversely blur their boundaries.

4.1. L^1 Regularized Differentiation

The partial derivatives I_x and I_y of an image will be estimated using an anti-differentiation characterization. The method for I_x will be explained in this section. The derivative I_y is computed by the same scheme applied to the transposed image. Since only two images are available along the time axis, the temporal derivative will be estimated using the Horn and Schunck definition of the temporal derivative (Horn and Schunk, 1981). Let f designate an approximation of the derivative. Recall that in Mitiche et al. (2015), the derivative was computed by minimizing the following functional with respect to f :

$$E(f) = \frac{1}{2} \int_{\Omega} \left(\|Df - I\|^2 + \gamma \|\nabla f\|^2 \right) dx dy, \quad (13)$$

where D is the anti-differentiation operator, γ is a positive constant, and ∇_f is the spatial gradient of f . The integral operator of anti-differentiation D is defined by

$$Df(x, y) = \int_0^x f(z, y) dz \quad (14)$$

To generalize this formulation to preserve boundaries, we will replace the L^2 regularization term in equation (13) by an L^1 term: $\int_{\Omega} (f_x^2 + f_y^2)^{\frac{1}{2}} dx dy$. The objective functional becomes

$$E(f) = \frac{1}{2} \int_{\Omega} \left(\|Df - I\|^2 + \gamma \int_{\Omega} (f_x^2 + f_y^2)^{\frac{1}{2}} \right) dx dy, \quad (15)$$

The corresponding Euler–Lagrange equations are

$$D^*(Df - I) - \gamma \frac{\partial}{\partial x} \frac{f_x}{(f_x^2 + f_y^2)^{\frac{1}{2}}} - \gamma \frac{\partial}{\partial y} \frac{f_y}{(f_x^2 + f_y^2)^{\frac{1}{2}}} = 0, \quad (16)$$

where D^* is the adjoint operator of D given by, assuming $\Omega = [0, l] \times [0, l]$, $D^*f(x, y) = \int_x^l f(z, y) dz$.

Non-linearity occurs in the regularization terms of equation (23). As done in the previous section, we can, in practice, and without affecting in any significant way subsequent processing, extend differentiability to the origin by replacing the denominators by $(f_x^2 + f_y^2 + \varepsilon)^{\frac{1}{2}}$, for some small positive ε . Also, and as we have done in the previous section, solve equation (16) iteratively, by evaluating, at each iteration k , the non-linear terms at the preceding iteration $k - 1$. More precisely, we have, after initialization, the following equation at iteration k :

$$D^*(Df^k - I) - \frac{\gamma}{\left((f_x^{k-1})^2 + (f_y^{k-1})^2 + \varepsilon \right)^{\frac{1}{2}}} \nabla^2 f^k = 0 \quad (17)$$

As in Mitiche et al. (2015), discretization of equation (17) yields a large-scale sparse system of linear equations which can be solved by the Gauss–Seidel method.

4.2. Example

We apply the scheme to compute the derivatives of the noisy (Gaussian white noise, $SNR = 1$) synthetic *chessboard* image of **Figure 2**, which illustrates that L^1 regularized differentiation outperforms both the local finite-difference definition and L^2 regularized differentiation. A quantitative evaluation can be done by computing the MSE (mean squared error) and the SDE (standard deviation of error). **Table 2** lists the results. Derivatives are in gray level (0–255 range) per pixel. The top part of the table gives the measurements as obtained from the whole image. The values listed in the bottom part of the table come from the vicinity of the image boundaries, using a 5×5 window centered on these. The results confirm visual inspection, i.e., that L^1 regularization outperforms both L^2 regularization and local finite differences.

5. EXPERIMENTAL RESULTS

In this section, we expose various experiments showing the application of the described method to synthetic and real image sequences. In addition to displaying various experimental examples, we present a comparative and quantitative analysis that highlights the positive effects of L^1 regularization over the whole image domain, and particularly within motion boundaries. In our experiments, all formulation parameters are determined empirically and distances are measured in pixels. In all the examples, regularized differentiation's coefficient γ has been fixed equal to 1. As approximated in Sekkati and Mitiche (2007), the focal length of the camera is 600 pixels. The position of the frontoparallel plane Z_0 is fixed to 6×10^4 pixels. The initial values of scene flow and depth at each point are set to, respectively, 0 and Z_0 . Coefficients α and β vary from a sequence to another and are given in the caption of each example.

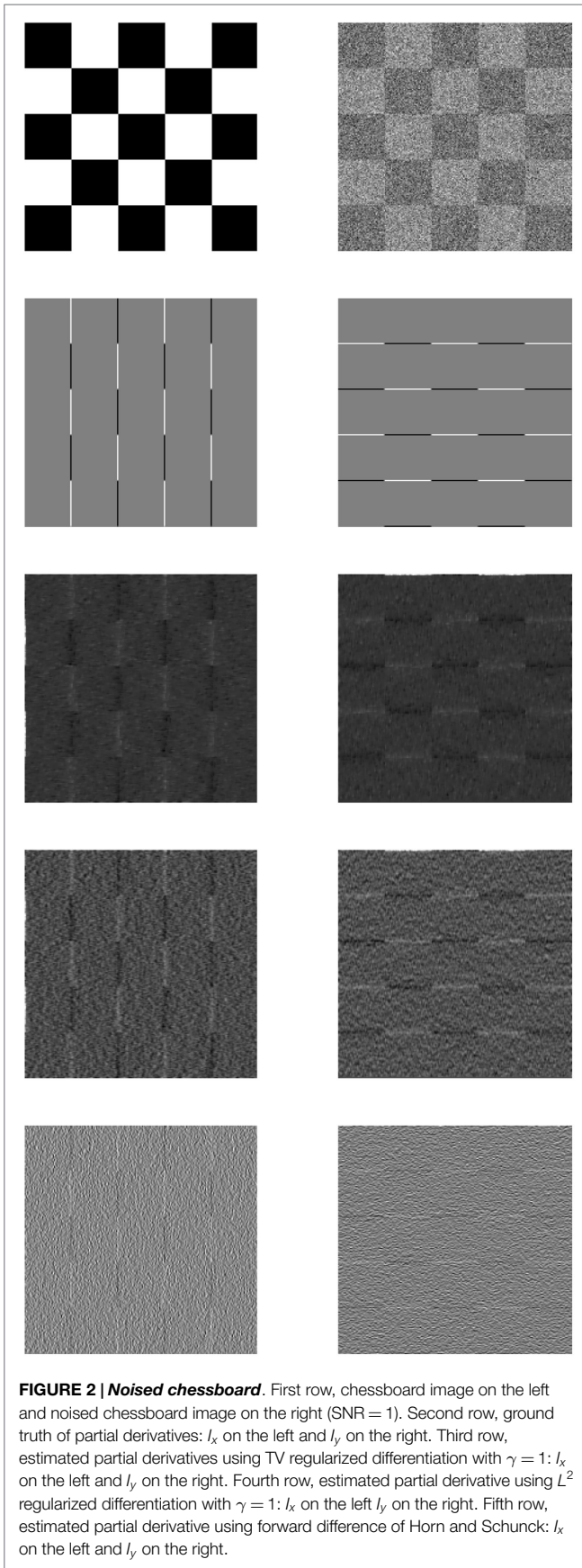


TABLE 2 | Differentiation: L^1 regularization, L^2 regularization, and local finite differences (LFD) applied to noised chessboard image (SNR = 1) and evaluated using mean squared error (MSE) and standard deviation error (SDE).

Methods	L^1	L^2	LFD
Noised chessboard image	MSE = 0.06 SDE = 0.09	MSE = 0.14 SDE = 0.13	MSE = 0.41 SDE = 0.31
Noised chessboard boundaries	MSE = 0.02 SDE = 0.07	MSE = 0.03 SDE = 0.09	MSE = 0.06 SDE = 0.16

Top: measurements from the whole image. Bottom: values from 5×5 windows centered on the boundaries. Regularization coefficient $\gamma = 1$. Bold font indicates the best values.

5.1. Examples

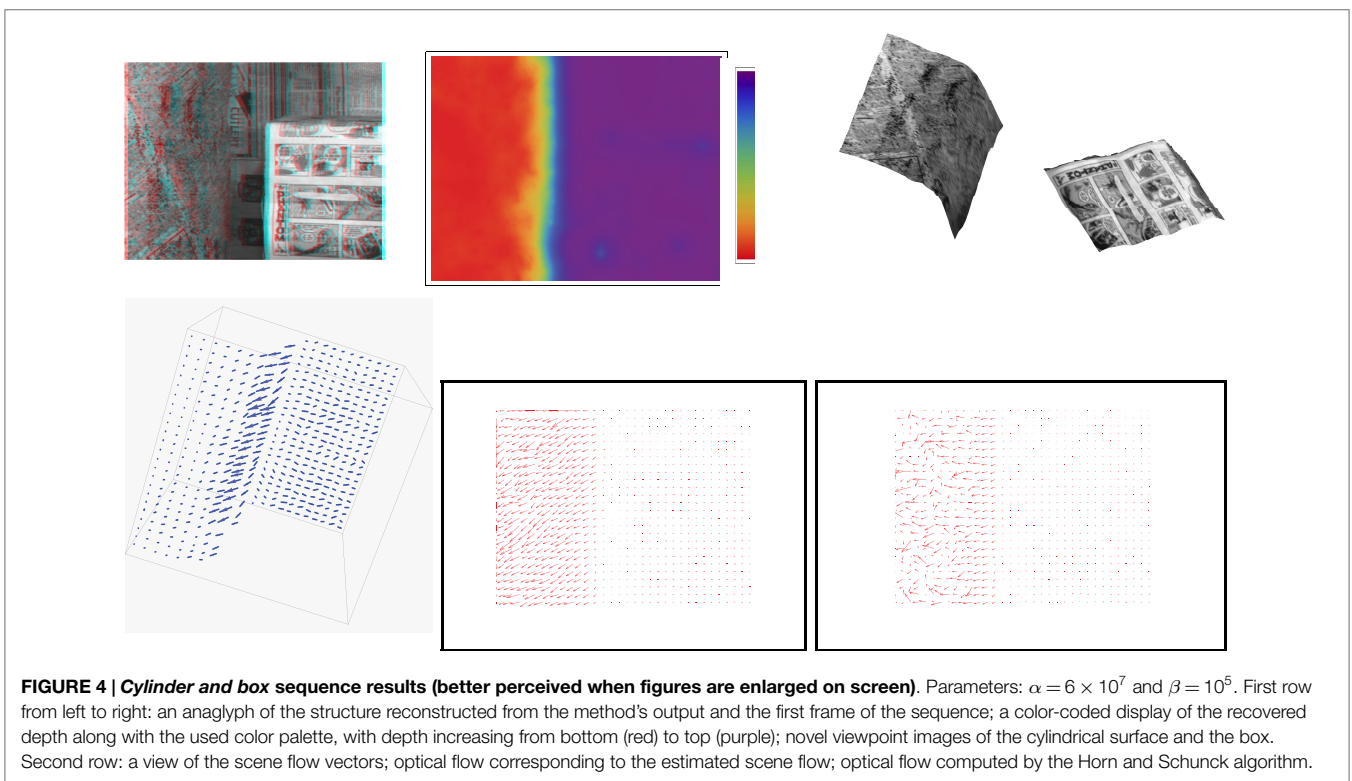
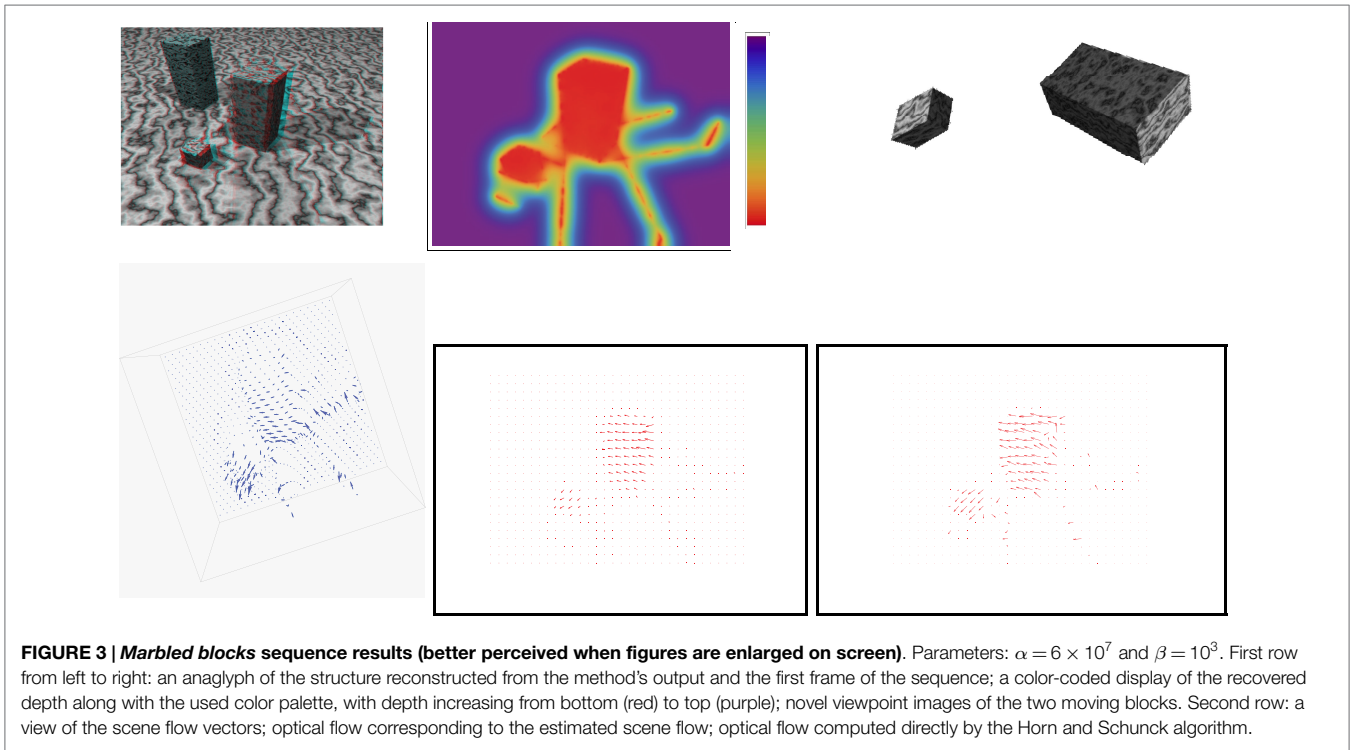
Four image sequences with different characteristics served as samples to test the validity of our scheme and its implementation. Our displays include

- Anaglyphs, which provide a convenient way for subjective appraisal of the recovered object depth. When viewed with chromatic (red–cyan) glasses on good-quality photographic paper or on standard screens, anaglyphs give viewers a strong sense of depth. They are produced from one of the two input images and the computed depth map and are generally better perceived with full color resolution;
- Standard displays using color-coded depth so as to highlight image-depth variations;
- 3D reconstructed objects;
- 3D scene flow vector fields; and
- 2D optical flow fields corresponding to our recovered scene flow. This can serve as an indirect validation of our implementation when the 2D outputs are compared to standard optical flow methods (e.g., the well tested/researched benchmark algorithm of Horn and Schunck).

We provide several figures, each corresponding to an example and organized as follows. The first row includes (from left to right): (a) an anaglyph of the scene structure reconstructed from our method's output and the first frame of the sequence; (b) a color-coded display of the recovered depth and the used color palette, with depth increasing from bottom (red) to top (purple); and (c) novel viewpoint images of the moving objects in the scene. The second row depicts (from left to right): (a) a view of the obtained scene flow; (b) a projected optical flow corresponding to our estimated scene flow; and (c) optical flow computed directly by the Horn and Schunck algorithm.

The following describes the four image sequences that we used:

- The *Marbled-block* synthetic image sequence (**Figure 3**) is taken from the database of KOGS/IAKS Laboratory, Germany. This sequence shows three blocks, two of them are moving. The rightmost block is moving backward to the left, whereas the front-most (smallest) block moves forward to the left. Some aspects of this sequence make its 3D interpretation challenging: the blocks and the floor have a similar macrotecture with weak spatiotemporal intensity variations within the textons. This makes the occluding boundaries of the blocks ill defined at



some places. Also, the source of light position with respect to the blocks causes shadows, which move with the blocks.

- The *Cylinder and box* real sequence (Figure 4), provided in Debrunner and Ahuja (1998), depicts two moving objects,

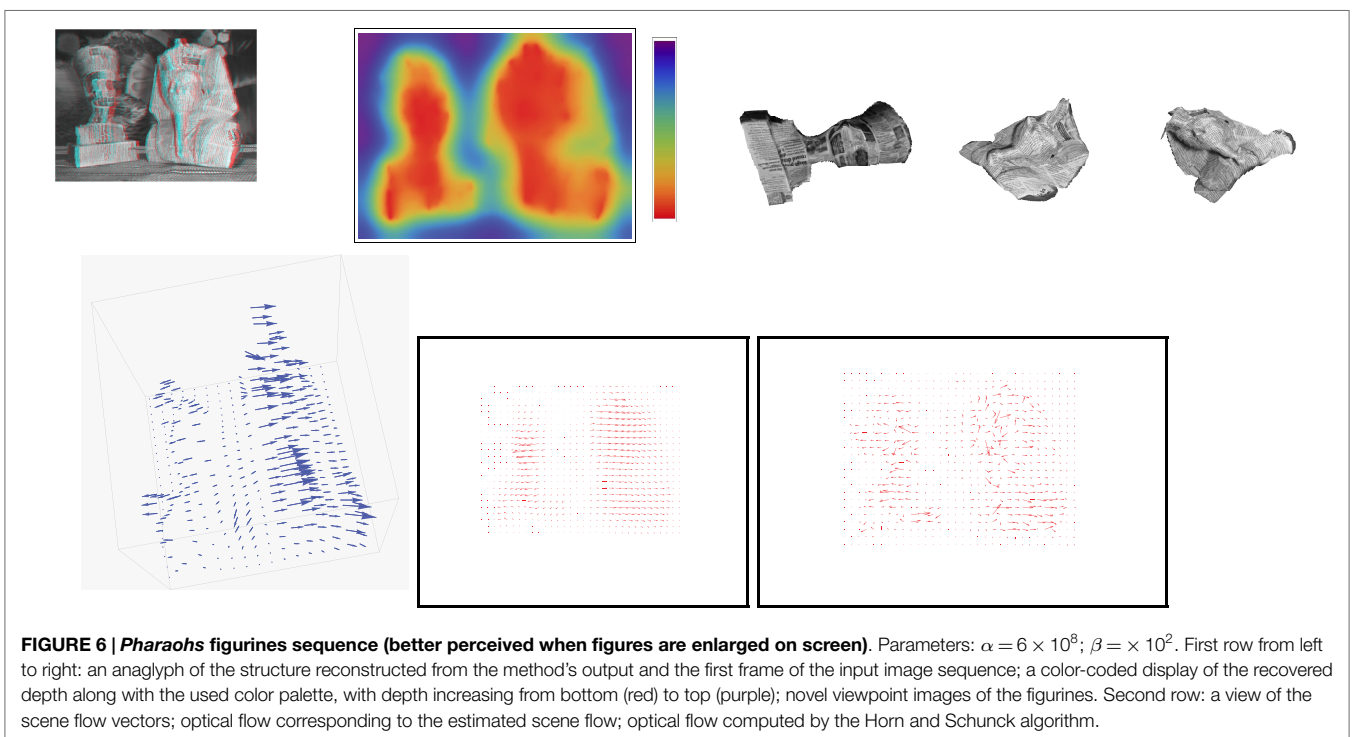
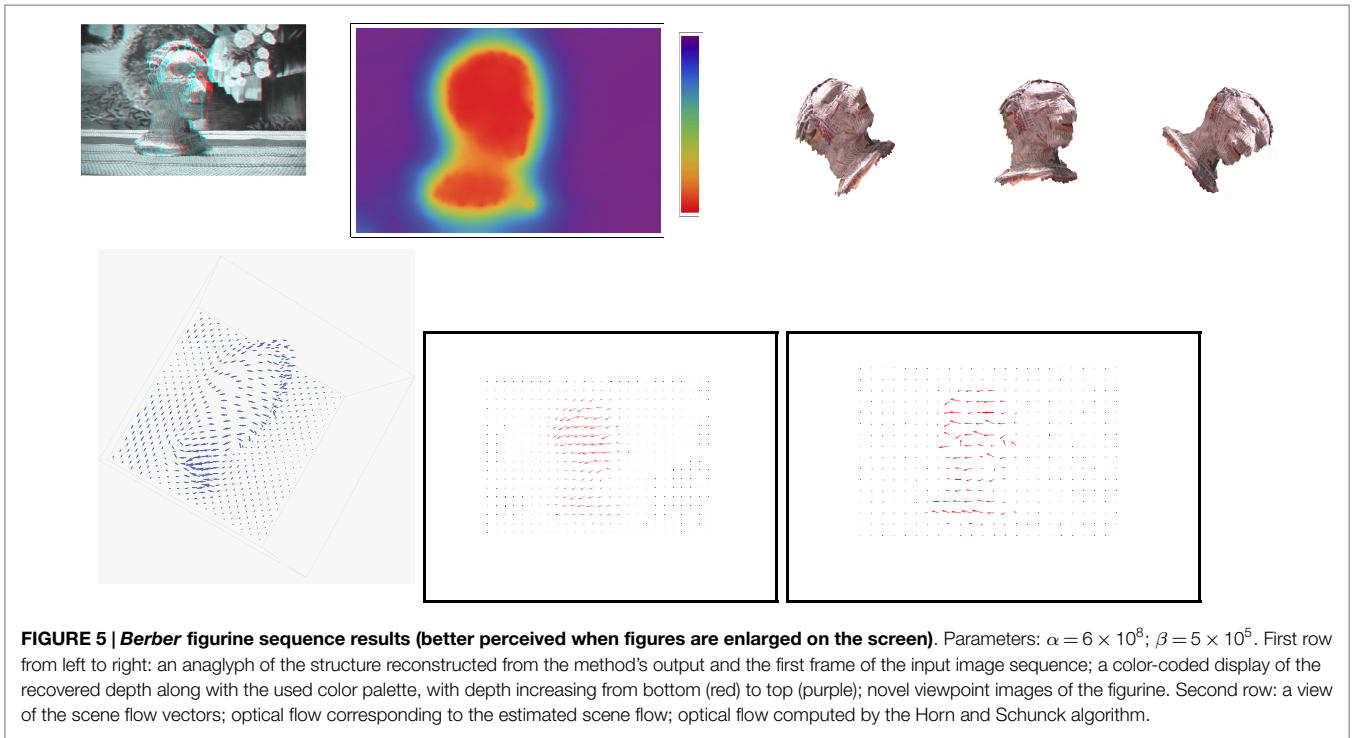
along with a moving background: a cylindrical surface rotating at a velocity of 1° per frame about the vertical axis and moving laterally at an image rate of about 0.15 pixel per frame toward the right, as well as a box translating at approximately 0.30 pixel

per frame toward the right. Also, the background is translating to the right (parallel to the box motion) at a rate of about 0.15 pixel per frame. Those motions make 3D interpretation and recovery challenging in this example.

- The *Berber* real sequence (**Figure 5**) exhibits a sculpture rotating about the vertical axis and translating forward to the left in a static environment.

- The *Pharaohs* sequence (**Figure 6**) shows two moving sculptures in a static environment. The first (leftmost) figurine translates left and forward, whereas the second rotates about a nearly vertical axis to the right.

The examples above support the validity of our scheme and its implementation. The obtained anaglyphs, color-coded depths,

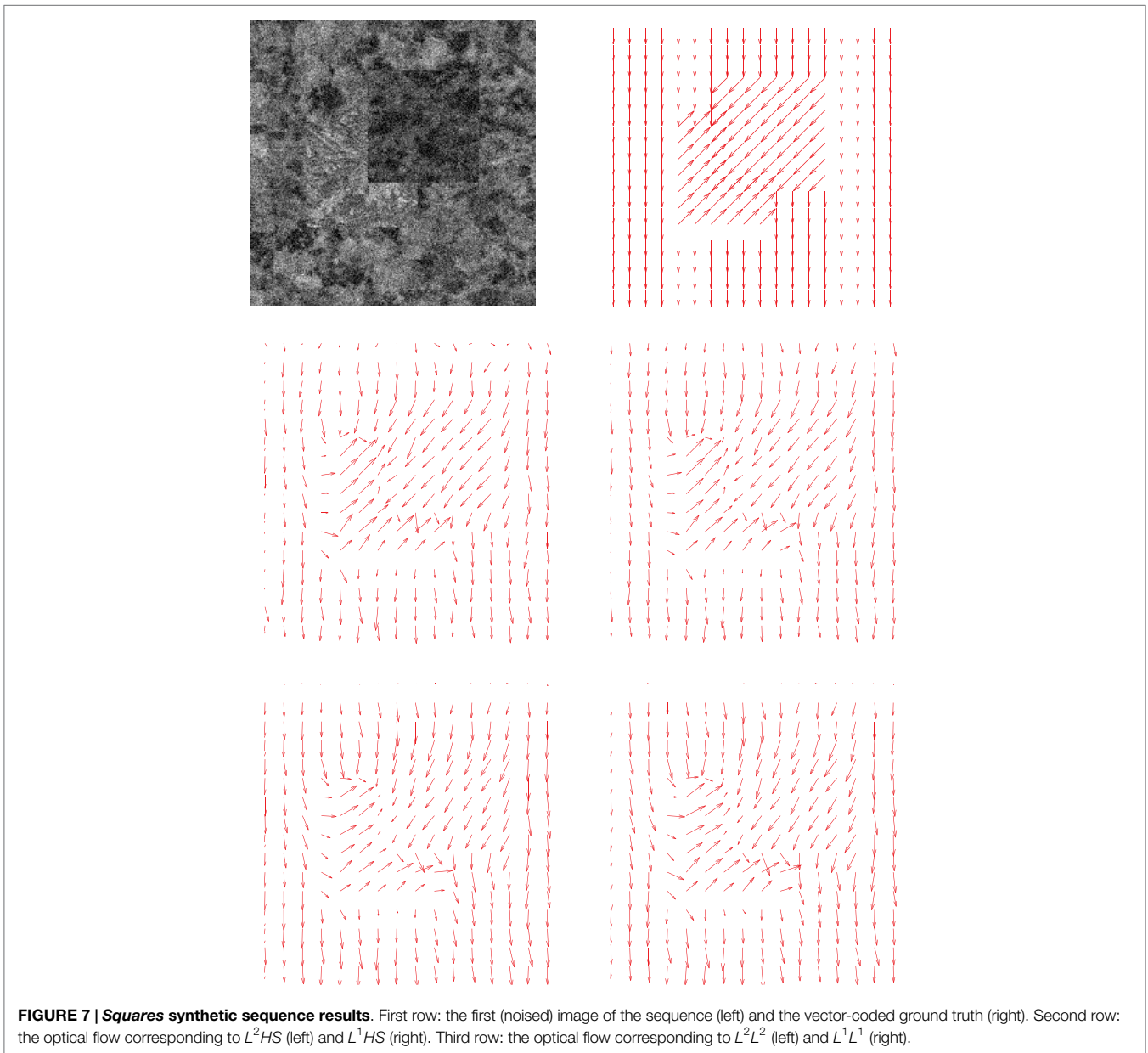


and 3D object reconstructions are consistent with the actual structures of the scenes. The displays of the estimated 3D scene flow and the corresponding 2D optical flow are consistent with the real motion of the objects in scenes. Also, the optical flow derived from our scene flow estimation is in line with a direct optical flow computation by the standard Horn and Schunck algorithm. We notice that the obtained fields are more regular and present less noise than those computed directly with Horn and Schunck algorithm. This is mainly due to the use of 3D information and regularized differentiation.

5.2. Comparative Analysis

We report a comprehensive comparative analysis, which demonstrates the benefits of our L^1 formulations. In particular, we focus our evaluations on the boundaries of motions and objects, so as to illustrate the boundary-preserving effect of our L^1 methods.

We began our comparative analysis by the simple synthetic sequence *Squares*, which includes two images with known motions. This sequence depicts two overlapping squares in opposite motions, along with a moving background. The motions for these three elements are known: a translation of the rightmost square by $(-1, -1)$ pixels in the downward-left direction, a translation of the leftmost square by $(1, 1)$ pixels in the top-right direction, and downward translation of the background by $(0, -1)$ pixels. To better test the performance of evaluated schemes, we added noise independently to the first and second image. Noise values are from a discretized, shifted, and truncated Gaussian in the interval between 0 and 100 gray levels, within an overall range of $[0, 255]$. The first row of **Figure 7** displays the first (noised) image (left) and the vector-coded ground truth (right). The second row depicts the optical flows, the first (left) obtained from a projection of the L^2 regularized scene flow (Mitiche et al.,



2015) and the second from our L^1 regularized scene flow. In both cases, we estimated image derivatives with a finite-difference regularization based on the standard Horn and Schunck definition (Horn and Schunck, 1981). We refer to these methods as L^2HS (left) and L^1HS (right). In the third row, we repeated the same experiment using regularized image differentiation instead of the finite differences. On the left, we depict the result using L^2 regularization for both scene flow and image-derivative estimations (L^2L^2) whereas on the right, we show the result for L^1 (L^1L^1). The optical flows resulting from the four methods are consistent with their expected overall appearance. However, we see clearly a difference between the performances of those algorithms.

Visually, the L^1L^1 scheme yielded the closest match to the vector-coded ground truth. To support this quantitatively, we added an evaluation based on two standard error measures for optical flow (Baker et al., 2011): average angular error (aae) and

endpoint error (epe); see Table 3. The (L^1L^1) scheme performed better than the other methods.

Figure 8 depicts another example for our comparative analysis. It uses the *Hydrangea* sequence of a real scene from the Middlebury data set (Baker et al., 2011). The sequence shows a rotating flower bouquet within a translating background, and

TABLE 3 | Performance of L^2HS , L^1HS , L^2L^2 , and L^1L^1 algorithms on the noised squares image (SNR = 1.12).

L^2HS	L^1HS	L^2L^2	L^1L^1
aae = 15.94	aae = 12.57	aae = 15	aae = 11.95
epe = 0.44	epe = 0.41	epe = 0.4	epe = 0.36

Bold font indicates the best values.

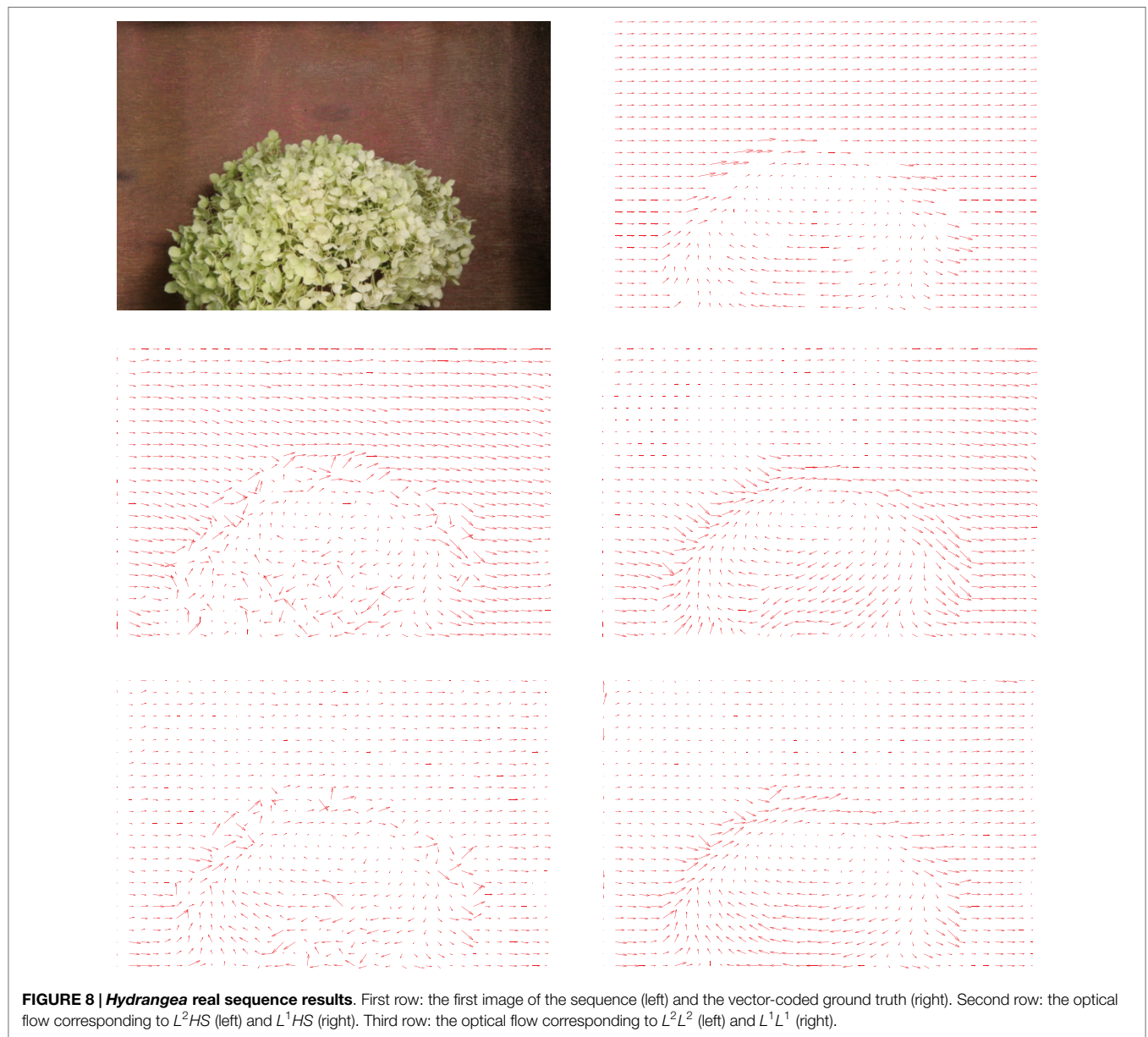


FIGURE 8 | *Hydrangea* real sequence results. First row: the first image of the sequence (left) and the vector-coded ground truth (right). Second row: the optical flow corresponding to L^2HS (left) and L^1HS (right). Third row: the optical flow corresponding to L^2L^2 (left) and L^1L^1 (right).

the ground-truth flow of the sequence is given.¹ The first row of the figure displays the first of the two input images (left) and the vector-coded ground truth (right). The second row depicts the optical flows, the first (left) obtained from a projection of the L^2 regularized scene flow (Mitiche et al., 2015) and the second from our L^1 regularized scene flow. In both cases, we estimated image derivatives with a finite-difference regularization based on the standard Horn and Schunck definition (Horn and Schunk, 1981). We refer to these methods as L^2HS (left) and L^1HS (right). In the third row, we repeated the same experiment using regularized image differentiation instead of the finite differences. On the left,

we depict the result using L^2 regularization for both scene flow and image-derivative estimations (L^2L^2) whereas on the right, we show the result for L^1 (L^1L^1). Visually, the L^1L^1 scheme yielded the closest match to the vector-coded ground truth. **Table 4** supports

TABLE 4 | Performance of L^2HS , L^1HS , L^2L^2 , and L^1L^1 algorithms on the *Hydrangea* real sequence.

L^2HS	L^1HS	L^2L^2	L^1L^1
aae = 21.18	aae = 16.72	aae = 17.04	aae = 15.96
epe = 2.17	epe = 1.78	epe = 1.92	epe = 1.54

Bold font indicates the best values.

¹<http://vision.middlebury.edu/flow/>

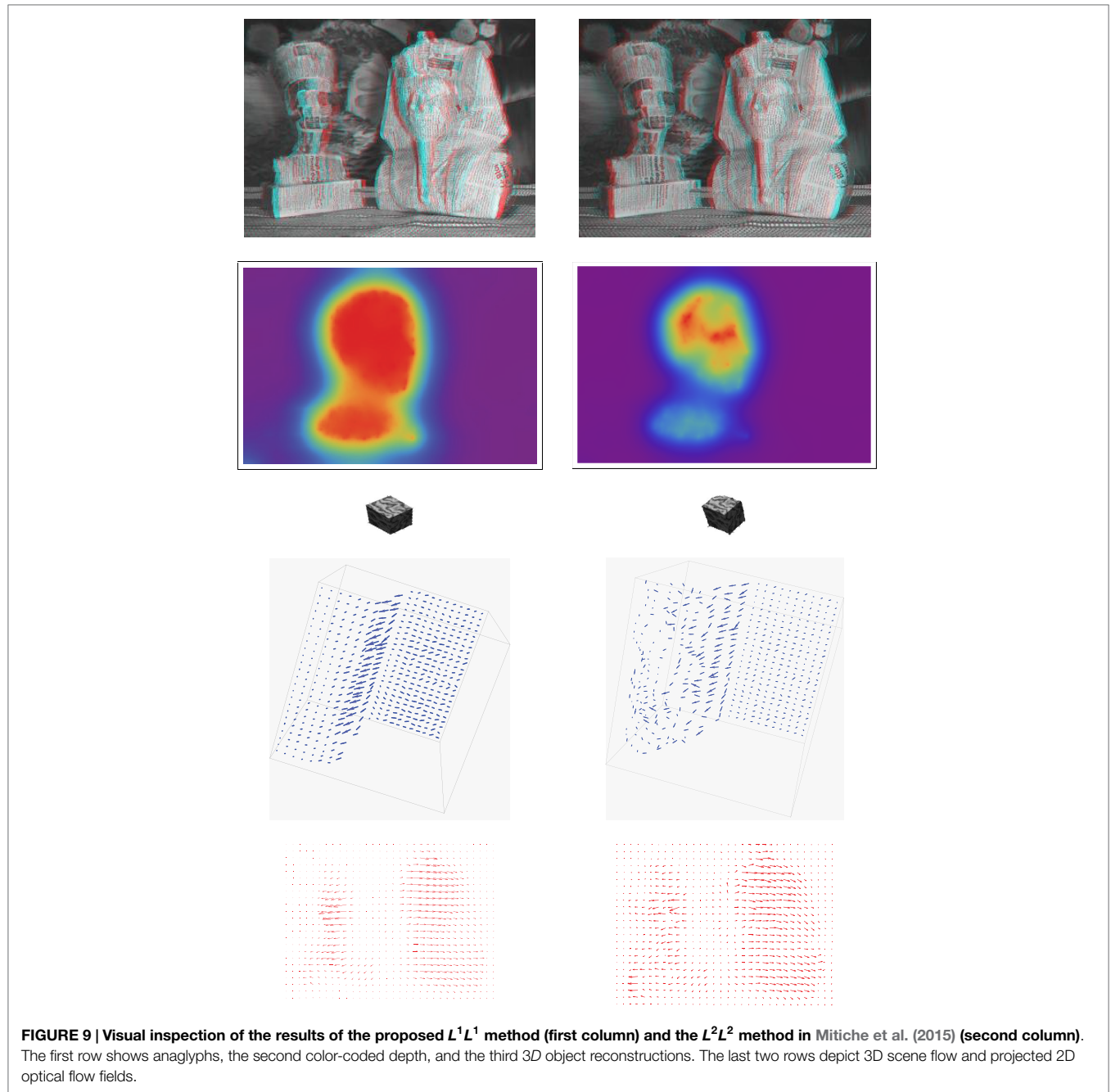


FIGURE 9 | Visual inspection of the results of the proposed L^1L^1 method (first column) and the L^2L^2 method in Mitiche et al. (2015) (second column). The first row shows anaglyphs, the second color-coded depth, and the third 3D object reconstructions. The last two rows depict 3D scene flow and projected 2D optical flow fields.

quantitatively these results. It reports two standard optical flow errors (aae and epe) (Baker et al., 2011), showing that the L^1L^1 scheme performed better than the other methods.

Figure 9 compares visually the results of L^1L^1 (proposed method) and L^2L^2 (Mitiche et al., 2015) using the different examples in Section 1. The first column depicts the results of the proposed L^1L^1 method, whereas the second shows the results of Mitiche et al. (2015) (L^2L^2). The first row shows anaglyphs, the second color-coded depth, and the third 3D object reconstructions. The last two rows displayed both 3D scene flow and projected 2D optical flow fields. We can see that the results are better with L^1L^1 than with L^2L^2 : the 3D parameters are better defined, clearer, and sharper, especially on the boundaries; flow fields are more regular and smooth.

Tables 5 and 6 report quantitative comparisons of the four algorithms (L^2HS and L^1HS in Table 5; L^2L^2 and L^1L^1 in Table 6) using three standard error measures (Baker et al., 2011): average angular error (aae), standard angular error (stae), and endpoint error (epe). Errors are computed between the motion resulting from scene flow and the optical flow ground truth. As we do not have scene flow ground truth for these real examples, evaluations of the resulting (projected) optical flow are a good indirect way to assess scene flow results. We constructed an optical flow ground truth using SURF-based (Bay et al., 2008) detection and correspondences of the key points in the two images. Then, the velocity coordinates of these key points are computed, yielding an optical flow ground truth. Tables 5 and 6 confirm that the use of our L^1 regularization improves the results and that the L^1L^1 formulation outperforms the other methods.

In the following part of our comparative analysis, we will focus on the assessment (both qualitative and quantitative) of motion within objects boundaries, as preserving these is an important feature of our L^1 formulation.

Let us start with a qualitative visual inspection by displaying the images of the gradient of motion for each of the four methods (L^2HS , L^1HS , L^2L^2 , and L^1L^1); see Figure 10. We note that with our L^1 regularization, points on the boundaries of motion are brighter (sharper). This is due to the fact that our L^1 formulation preserves sharp boundaries.

To support these results, we added quantitative evaluations based on aae, stae, and epe errors: Table 7 reports the results for L^2HS and L^1HS ; Table 8 reports the results for L^2L^2 and L^1L^1 . Errors are computed between the motion projected from the obtained scene flow and the optical flow ground truth within 7×7 windows centered at a set of key points on motion boundaries (the key points are those for which we have motion ground truth). Tables 7 and 8 report the results, which clearly indicate that our L^1L^1 formulation outperforms all the other methods.

6. CONCLUSION

This study investigated a boundary-preserving method for joint recovery of scene flow and relative depth from a monocular sequence of images. The scheme built upon the basic formulation of Mitiche et al. (2015). It minimized a functional composed of the data conformity term of Mitiche et al. (2015), which relates

TABLE 5 | Errors for the L^2HS and L^1HS formulations.

Sequences	Errors	L^2HS	L^1HS
Berber	aae	38.51	22.01
	stae	43.84	24.18
	epe	1.11	0.68
Pharaohs	aae	59.17	43.99
	stae	41.38	30.37
	epe	1.12	1.54
Cylinder	aae	59.26	57.7
	stae	48.53	46.82
	epe	2.03	2.03
Marbled-block	aae	8.89	7.63
	stae	22.79	17.55
	epe	0.2	0.15

Bold font indicates the best values.

TABLE 6 | Errors for the L^2L^2 and L^1L^1 formulations.

Sequences	Errors	L^2L^2	L^1L^1
Berber	aae	23.27	11.61
	stae	14.77	10.16
	epe	0.9	0.41
Pharaohs	aae	20.01	14.02
	stae	19.21	12.04
	epe	0.62	0.36
Cylinder	aae	18.68	10.05
	stae	18.91	9.55
	epe	0.78	0.2
Marbled-block	aae	4.14	4.2
	stae	8.56	8.33
	epe	0.1	0.09

Bold font indicates the best values.

the image sequence spatiotemporal variations to scene flow and depth, and an L^1 regularization term, rather than L^2 as in Mitiche et al. (2015). Therefore, this afforded a boundary-preserving version of the basic formulation. The corresponding non-linear Euler–Lagrange equations were discretized and solved iteratively by a scheme, which solved at each iteration a large-scale sparse system of linear equations in the unknowns of scene flow and depth. The image derivatives were estimated by a variational method with L^1 regularization. This also led to an iterative method of resolution, which consisted of solving a large sparse system of linear equations at each iteration. Experiments show that the scheme is sound and efficient. The examples demonstrated the need to regularize scene flow and depth so as to take into account their boundaries, i.e., sharp spatial transitions of scene flow or depth. The results justify extensive further investigations, particularly concerning quantitative, i.e., ground truth controlled evaluation, motion of large extent, and image noise and resolution in common practical settings.

The choice of a continuous alternating optimization scheme for our problem can be motivated by two important facts. First, we are dealing with continuous variables, and therefore, a continuous (not discrete) Euler–Lagrange regularization approach is a natural choice. Furthermore, our alternating scheme for solving the ensuing non-linear Euler–Lagrange equations has a computational

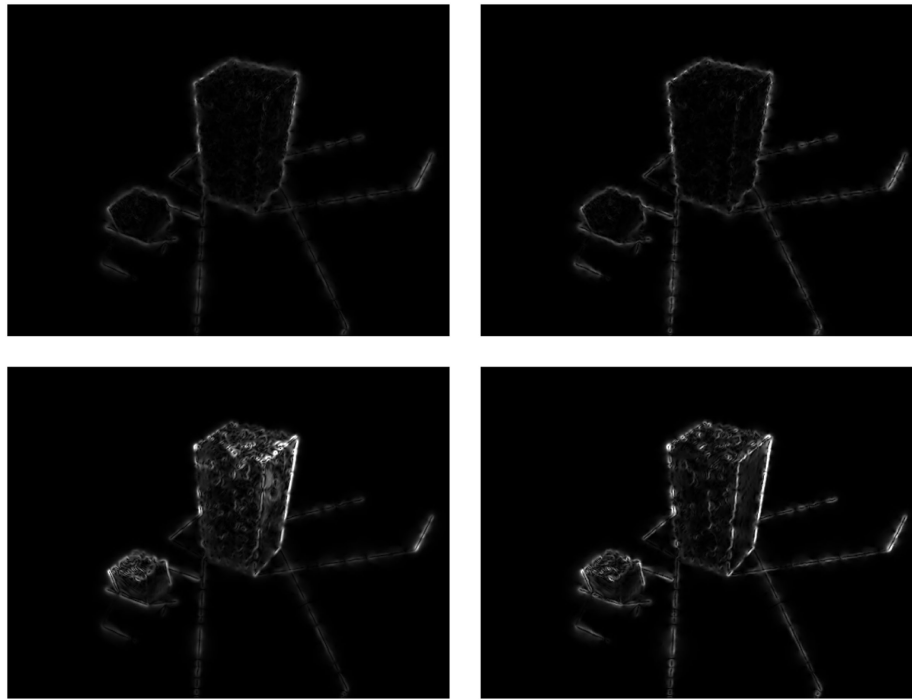


FIGURE 10 | Gradients of optical flow for the Marbled blocks sequence. First row: $L^2 HS$ (left) and $L^1 HS$ (right). Second row: $L^2 L^2$ (left) and $L^1 L^1$ (right).

TABLE 7 | Quantitative evaluations on the boundaries of motion ($L^2 HS$ and $L^1 HS$).

Sequences	Errors	$L^2 HS$	$L^1 HS$
Berber ($Pt = 14$)	aae	51.4	47.24
	stae	23.47	21.72
	epe	1.53	1.43
Pharaohs ($Pt = 29$)	aae	45.49	32.23
	stae	28.73	18.21
	epe	0.97	0.69
Cylinder ($Pt = 15$)	aae	24.78	23.45
	stae	13.75	13.36
	epe	0.46	0.44
Marbled-block ($Pt = 9$)	aae	94.35	78.81
	stae	15.33	7.07
	epe	2.32	1.82

Bold font indicates the best values.

complexity that behaves linearly w.r.t the number of grid points (N). This is important in practice, particularly when dealing with large image sequences. Of course, it would be very interesting to investigate other regularization options for estimating scene flow and depth from a single image sequence, for instance:

- Discrete Markov Random Fields (MRFs): MRFs were previously investigated for optical flow using sub-modular pairwise potentials (Lempitsky et al., 2008). MRF models can benefit from powerful combinatorial optimization techniques such as graph cuts (Boykov et al., 2001). It is worth noting, however, that adapting discrete MRFs to our continuous setting requires some technical care and is not straightforward.

TABLE 8 | Quantitative evaluations on the boundaries of motion ($L^2 L^2$ and $L^1 L^1$).

Sequences	Errors	$L^2 L^2$	$L^1 L^1$
Berber ($Pt = 14$)	aae	26.02	23.27
	stae	20.54	14.77
	epe	1.01	0.9
Pharaohs ($Pt = 29$)	aae	15.45	14.02
	stae	11	12.04
	epe	0.36	0.36
Cylinder ($Pt = 15$)	aae	11.99	10.05
	stae	9.66	9.55
	epe	0.23	0.2
Marbled-block ($Pt = 9$)	aae	20.44	11.53
	stae	11.84	6.24
	epe	0.98	0.62

Bold font indicates the best values.

- Regularization based on non-local means filtering (Favaro, 2010). Non-local means can preserve edges and textures. They were applied successfully to thin structures in depth-from-defocus problems (Favaro, 2010).
- State-of-the-art solvers for non-smooth problems such as the primal-dual algorithm of Chambolle and Pock (2011).

AUTHOR CONTRIBUTIONS

Ms. YM is a PhD student at INRS under the supervision of Prof. AM. Dr. IA is a professor at École de Technologie Supérieure, Montreal, and co-supervisor of YM. The authors have all contributed, to varying degrees, to the theoretical/methodological/experimental aspects of this paper.

REFERENCES

- 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
- Aubert, G., and Kornprobst, P. (2002). *Mathematical Problems in Image Processing*. Springer.
- Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* 92, 1–31. doi:10.1007/s11263-010-0390-2
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vision Image Understand.* 110, 346–359. doi:10.1016/j.cviu.2007.09.014
- Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 1222–1239. doi:10.1109/34.969114
- Chambolle, A., and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vision* 40, 120–145. doi:10.1007/s10851-010-0251-1
- Ciarlet, P. (1994). *Introduction à l'analyse numérique matricielle et à l'optimisation*, 5th Edn. Masson.
- 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
- Favaro, P. (2010). “Recovering thin structures via nonlocal-means regularization with application to depth from defocus,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1133–1140.
- Forsythe, G., Malcolm, M., and Moler, C. (1977). *Computer Methods for Mathematical Computations*. Prentice-Hall.
- Heinrich, M. P., Jenkinson, M., Bhushan, M., Matin, T., Gleeson, F. V., Brady, M., et al. (2012). Mind: modality independent neighbourhood descriptor for multimodal deformable registration. *Med. Image Anal.* 16, 1423–1435. doi:10.1016/j.media.2012.05.008
- Horn, B., and Schunk, B. (1981). Determining optical flow. *Artif. Intell.* 17, 185–203. doi:10.1016/0004-3702(81)90024-2
- Huguet, F., and Devernay, F. (2007). “A variational method for scene flow estimation from stereo sequences,” in *2007 IEEE 11th International Conference on Computer Vision (ICCV)* (IEEE), 1–7.
- Lempitsky, V. S., Roth, S., and Rother, C. (2008). “FusionFlow: discrete-continuous optimization for optical flow estimation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.
- Longuet-Higgins, H., and Prazdny, K. (1981). The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B* 208, 385–397. doi:10.1098/rspb.1980.0057
- Marshall, S. (2005). Depth dependence of ambient noise. *IEEE J. Oceanic Eng.* 30, 275–281. doi:10.1109/JOE.2005.850876
- Mitiche, A., and Aggarwal, J. (2013). *Computer Vision Analysis of Image Motion by Variational Methods*. Springer.
- 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
- 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
- Periaswamy, S., and Farid, H. (2006). Medical image registration with partial data. *Med. Image Anal.* 10, 452–464. doi:10.1016/j.media.2005.03.006
- Pons, J.-P., Keriven, R., and Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vis.* 72, 179–193. doi:10.1007/s11263-006-8671-5
- Sekkati, H., and Mitiche, A. (2007). A variational method for the recovery of dense 3D structure from motion. *J. Rob. Auton. Syst.* 55, 597–607. doi:10.1016/j.robot.2006.11.006
- Stoer, J., and Bulirsch, R. (2002). *Introduction to Numerical Analysis, Texts in Applied Mathematics*, 3rd Edn, Vol. 12. New York: Springer.
- Sun, D., Roth, S., and Black, M. J. (2010). “Secrets of optical flow estimation and their principles,” in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 2432–2439.
- Vedula, 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)*.
- Vogel, C. R. (2002). *Computational Methods for Inverse Problems*. SIAM Frontiers in Applied Mathematics.
- Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., and Cremers, D. (2008). *Efficient Dense Scene Flow from Sparse or Dense Stereo Data*. Springer.

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.

Copyright © 2016 Mathlouthi, Mitiche and Ben Ayed. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.