Scene Flow from Depth and Color Images

Antoine Letouzey, Benjamin Petit, Edmond Boyer

Scene Flow from Depth and Color Images

Antoine Letouzey
antoine.letouzey@inria.fr
Benjamin Petit
benjamin.petit@inria.fr
Edmond Boyer
edmond.boyer@inria.fr

Morpheo Team
LJK / INRIA Grenoble Rhônes-Alpes
Grenoble, France

Abstract

In this paper we consider the problem of estimating a 3D motion field using multiple cameras. In particular, we focus on the situation where a depth camera and one or more color cameras are available, a common situation with recent composite sensors such as the Kinect. In this case, geometric information from depth maps can be combined with intensity variations in color images in order to estimate smooth and dense 3D motion fields. We propose a unified framework for this purpose, that can handle both arbitrary large motions and sub-pixel displacements. The estimation is cast as a linear optimization problem that can be solved very efficiently. The novelty with respect to existing scene flow approaches is that it takes advantage of the geometric information provided by the depth camera to define a surface domain over which photometric constraints can be consistently integrated in 3D. Experiments on real and synthetic data provide both qualitative and quantitative results that demonstrate the interest of the approach.

1 Introduction

Motion is an important feature when analyzing and interpreting dynamic real scenes. It provides rich and discriminative information on the objects that compose the scene and is used for instance by both human and artificial vision systems to track and segment such objects. The interest appears especially in interactive applications, *e.g.* games or smart environments, for which motion is a valuable source of information in the perception-action cycle. These applications often use depth cameras, *e.g.* time-of-flight or structured light cameras, that can directly perceive 3D locations without the need for additional multi-view processing. In this paper, we consider how to recover dense motion fields with such cameras.

Apart from active sensors or marker-based vision systems that can directly provide sparse motion information on moving scenes, less invasive methods traditionally consider intensity images and their temporal variations to derive estimates of dense motion fields. In the monocular case, 2D projections of such motion fields are estimated through the *optical flow* [6, 10]. When multiple cameras are available, the integration over different viewpoints allows a 3D motion field, the *scene flow* [11, 15], to be estimated. In both 2D and 3D cases, intensity variations alone are, however, not sufficient to estimate motion and additional constraints must be introduced, *e.g.* smoothness in most approaches. In this respect,

depth cameras advantageously provide useful geometric information from which additional consistent 3D smoothness constraints can be derived. However, few efforts have yet been made to integrate depth cameras in the estimation of motion fields. This is our objective in this paper. Specifically, we explore how instantaneous intensity variations can be used in combination with depth information to infer 3D motion information. To this purpose we build on a scheme recently proposed in the case of multiple color views [12] and we extend it to depth cameras. Such a scheme allows motion to be estimated without the need for spatial or temporal correspondences, as is required for tracking, though a few such correspondences can easily be integrated to improve the estimation. As shown in this paper, this makes it a simple and efficient way to obtain dense and instantaneous 3D motion information on moving objects using depth cameras.

The paper is organized as follows: in section 2 we review previous work. In section 3 we present the integration scheme. In section 4 we evaluate the approach both quantitatively and qualitatively, before discussing and concluding in section 5.

2 Related Work

A vast body of literature exists on how to recover motion fields using photometric information. Early works in this domain consider 2D motion fields between consecutive images and estimate them as the *optical flow* [1, 6] using the normal flow constraints derived from image intensity variations. When information from stereo images are available, 3D motion fields can be estimated. Such estimation can be achieved using known disparity information [15] or in combination with disparity estimation [7, 8]. In addition, temporal consistency can be enforced [14]. In the case of multiple image streams, both structure and motion can be estimated in a combined way [2, 13].

Closer to the configuration considered in this paper, some works assume a known 3D structure or shape, and consecutive images from one or several viewpoints. The estimated motion field, the *scene flow*, can then be obtained by projecting back onto the shape optical flow fields estimated in the images [15] or by directly integrating, onto the shape, the normal flow constraints coming from the images [11]. Of particular interest in this category is the work [12] that provides a scheme where various photometric constraints, normal flow and features, can be combined with a locally rigid deformation model. We extend this line of work to the configuration where a depth camera is available, hence providing structure information and therefore removing the need for structure estimation. To the best of our knowledge, this is the first attempt to estimate a 3D motion field using a depth camera and temporal intensity variations.

3 The Method

The proposed approach directly estimates a 3D motion field on the surface using 2D photometric constraints. To this aim, it takes as input a stream of depth and color images coming from calibrated and synchronized cameras. The associated camera configuration is a single depth camera combined with one or more color cameras. In the description below, and for the sake of simplicity, we will only consider a depth camera and a color camera. The extension to several color cameras is however straightforward, and will be discussed later in the

paper. Unlike previous work on surface flow [12] the proposed approach does not require a surface model to be determined, but takes advantage instead of the depth map and hence can consider a simple configuration with a depth camera and a single color camera, *e.g.* the Kinect camera.

3.1 Notation

At each time step t a color image I_t and a depth map D_t are acquired. We suppose that the cameras are calibrated and that projection matrices $\Pi_{cc}, \Pi_{dc} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ map 3D points $\mathbf{P} = (x,y,z)^T \in \mathbb{R}^3$ to 2D points $\mathbf{p} = (u,v)^T \in \mathbb{R}^2$ in the color image and the depth image respectively. The depth map yields a cloud of 3D points from which we can easily build a mesh surface S^t using the connectivity of the depth image, *i.e.* each pixel being a vertex in 3D is connected to its neighbors in the image.

From this input images, the approach estimates a dense 3D vector field V^t between time t and $t+1$ which corresponds to the set of instantaneous motion vector $V^t(\mathbf{P})$ of vertices in S^t (see figure 1). The optical flow v^t is then the projection of the estimated scene flow V^t onto the color image plane.

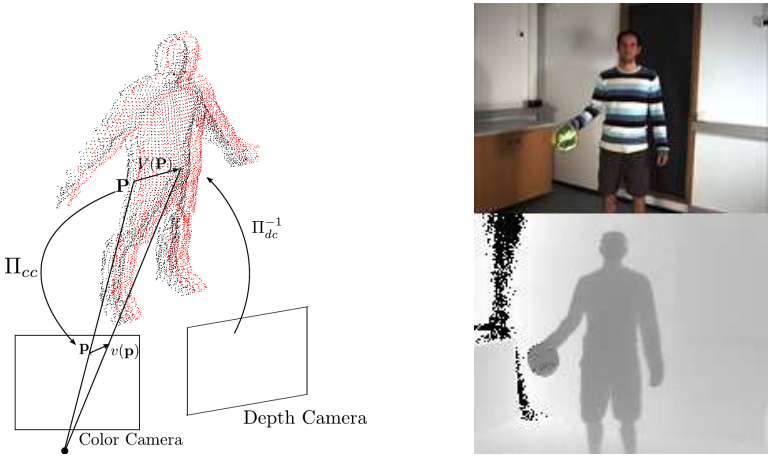


Figure 1: (a) Projection of the surface onto the color image plane, and (b) depth and color images.

In order to estimate the 3D flow $V^t(\mathbf{P})$, we cast the problem as an optimization where data terms corresponding to photometric consistency constraints are combined with a regularization term that favors smooth motion fields:

$$\mathbf{E} = \mathbf{E}_{data} + \mathbf{E}_{smooth}. \tag{1}$$

Data terms control both small and large displacements while the regularization term imposes a deformation model with local rigidity constraints. They are explained in the following.

3.2 Photometric Constraints

As suggested by Xu *et al.* in their work on optical flow [17], we used two different kinds of photometric information to deal with both large and small displacements. Each of these constraints adds a new component in the data term of the energy function (1).

3.2.1 Dense Normal Flow

Following traditional optical flow estimation, we assume brightness consistency between two consecutive projections of a 3D point on the same image plane. Though arguable, this assumption appears reasonable in our context where surfaces are often Lambertian and the illumination approximately constant for small displacements.

For infinitesimal displacements, we can then write the following *Normal Flow Equation* [1] for all pixels in I^t where the surface is visible:

$$\nabla I^t . v^t + \frac{\mathrm{d}I^t}{\mathrm{d}t} = 0 \ , \text{ or for 3D motion: } \nabla I^t . \left[J_\Pi V^t \right] + \frac{\mathrm{d}I^t}{\mathrm{d}t} = 0 \ , \tag{2}$$

where the associated 3D displacement of a point in the scene is related to the motion of its projection by the 2×3 Jacobian matrix $J_\Pi(\mathbf{p}) = \frac{\partial \mathbf{p}}{\partial \mathbf{P}}$. The equation (2) yields the following data term in (1):

$$\mathbf{E}_{flow} = \int_{I^t} \| \nabla I^t . \left[J_\Pi V^t \right] + \frac{\mathrm{d}I^t}{\mathrm{d}t} \|^2 \, \mathrm{d}\mathbf{p}. \tag{3}$$

This above criterion alone is not sufficient to estimate motion since the constraints are solely on the 2D motion in the direction normal to the gradient in the images; furthermore it is only valid for small displacements. This *aperture problem* in 2D is also true in 3D [15]. In addition, in 3D another limitation of this criterion results from the use of the Jacobian of the projection matrix, which is valid only for small displacements again. These limitations are addressed as follows.

3.2.2 Sparse Feature Correspondences

In order to handle large displacements in the scene we can consider another type of photometric consistency constraint. We choose to use a set of sparse feature correspondences between two consecutive color images I^t and I^{t+1} to help guide the motion field estimation. They can easily be extracted in the images using one of the numerous feature points detectors and descriptors, *e.g.* SIFT [9] or SURF [3]. These features act as anchor points for the motion estimation. The confidence we have in them is critical since outliers propagate error in the regularization process. We used SIFT features coupled with a conservative threshold on the confidence of the matching score (the same for all datasets under consideration) . This conservative strategy allowed us to remove all outliers in our experiments.

These 2D features directly constrain the optical flow, they can also constrain the scene flow by re-projecting them on the 3D surface. Thereby each pair of matched features gives a 3D-3D correspondence when the associated points are back-projected on the two consecutive surfaces, and thus a 3D displacement constraint, V_f. The associated data term writes:

$$\mathbf{E}_{3D} = \sum_{F^t} \| V^t - V_f^t \|^2 \ , \tag{4}$$

where F^t is the set of constrained 3D points corresponding to 2D matched features.

3.3 Motion Field Regularization

As mentioned earlier, normal flow constraints are not sufficient to estimate motion. Though providing additional information, 2D features only result in sparse local constraints. In order to estimate dense motion information, we use an additional assumption on the regularity of the motion field. With 2D optical flow, two main strategies can be followed: local or global regularization. Their extensions to 3D could both be considered but we chose a global regularization and, more specifically, the extension of the Horn and Schunck's method [6]. The reason is that while a global regularity is seldom true in images, especially at object boundaries, it often holds on surfaces. Thus, we use the following regularization term to ensure global smoothness of the estimated 3D vector field:

$$\mathbf{E}_{smooth} = \int_{S^t} \|\nabla V^t\|^2 \mathrm{d}\mathbf{P}. \tag{5}$$

Note that the above Laplacian smoothing term favors locally rigid deformations of the underlying mesh surface. In order to handle discontinuities of the surface, and as explained later, smoothness constraints can be weighted using the geometric information provided by the depth map.

3.4 Solving

By gathering all the terms previously defined, minimizing equation (1) is equivalent to solving the following Euler-Lagrange equation:

$$\lambda_{flow}^2 \left[\nabla I^t . \left[J_\Pi V^t \right] + \frac{\mathrm{d}I^t}{\mathrm{d}t} \right] + \lambda_{3D}^2 \delta_{F^t} \left[V^t - V_f^t \right] + \lambda_{smooth}^2 \nabla^2 V^t = 0, \tag{6}$$

where lambdas are scalar values used to weight the different terms and δ is the Kronecker symbol denoting that this term only involves a subset of points. This equations involves a set of linear constraints for each 3D point of the surface, therefore the minimization of the functional is equivalent to solving the following linear system:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{A} \end{bmatrix} V^t + \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} = 0, \tag{7}$$

where $\mathbf{L}$ is the graph Laplacian matrix of the surface mesh such that $\mathbf{L}(i,j)$ weights the relationship between point i and j (Laplacian weights are discussed in section 3.5.1) and $\mathbf{A}$ and $\mathbf{b}$ stack all the motion constraints coming from data terms. This linear system is very sparse and can be solved using any existing solver such as *Taucs*. Equation (6) may also be solved via an iterative scheme by applying the Jacobi method. We then solve the linear system at each point independently using the updated solution of the neighborhood.

3.5 Implementation Details

In this section some important choices made at the implementation level are detailed. First, weights that balance, in the regularization, the influence of mesh neighbors are discussed. Second, we explain how large and small displacements are separately handled in the approach.

3.5.1 Laplacian Weights

In the regularization term in expression (6), the continuous Laplace-Beltrami operator ∇^2 defined on the surface is approximated by the graph Laplacian matrix $\mathbf{L}$, *i.e.* $\nabla^2 V^t = \mathbf{L}V^t$ defined on the mesh, where:

$$\mathbf{L}(i,j) = \begin{cases} \deg(P_i) & \text{if } i = j, \\ -w_{ij} & \text{if } i \neq j \text{ and } P_i \text{ is adjacent to } P_j, \\ 0 & \text{otherwise ,} \end{cases}$$

with w_{ij} the edge weights and $\deg(P_i) = \sum_{j \neq i} w_{ij}$. The matrix $\mathbf{L}$ can be purely combinatorial, *i.e.* $w_{ij} \in \{0,1\}$, or involve weights $w_{ij} \geq 0$, *e.g.* cotangent weights, often used in computer graphics [16] with uniform sampling. In the context of this work, the mesh connectivity is taken from the image connectivity, *i.e.* neighboring points in the image are connected within the mesh representation. This yields a consistent mesh representation (*i.e.* without self-intersection) but with however potentially long edges corresponding to depth discontinuities. In order to properly handle these discontinuities in the regularization we propose the following weights:

$$w_{ij} = -G(|P_i - P_j|, \sigma),$$

where G denotes the Gaussian kernel, $|\cdot|$ is the Euclidean distance, and σ the standard deviation. In addition to strongly limit diffusion along long edges, Gaussian kernel weights have also been advocated by Belkin *et al.* [4] as providing good properties such as convergence to the continuous Laplace-Beltrami operator when increasing the mesh resolution.

3.5.2 Two Pass Algorithm

As explained before, normal flow information is only reliable at pixel scale and is not valid with large displacements. In the optical flow case, most existing methods rely on a multi-scale strategy to deal with large motion. Such a strategy is however not well suited in our context since it involves computing image pyramids, leading to image smoothing and hence 2D regularization that we wish to avoid in our approach. In the implementation, we perform the following two-step algorithm to overcome this problem:

- First, we only take into account the sparse feature correspondences and perform a preliminary estimation of V^t, solving equation (6) with $\lambda_{flow} = 0$. This step allows us to create a shifted surface $S^{t\prime}$ by applying the first estimation of V^t to S^t. We then re-project this surface into the color camera using texture from I^t.

- We now have two color images, one being the image of the re-projected surface and the other being I^{t+1}. They should be similar enough to compute reliable normal flow information. We can now combine large and small motion information in a second resolution of equation (6) with similar values for the different lambdas. In this second step, sparse features act as zero displacement anchor points instead of driving the deformation.

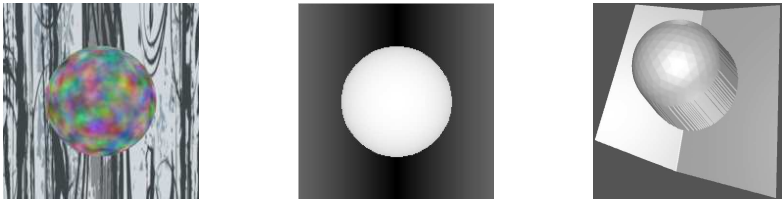
The final motion field is computed by combining the two estimations. In practice, the experiments show that if large motions are properly handled by the first step of the algorithm, the second one helps recovering smaller local details of the motion.

4 Evaluation

In order to evaluate the approach, we used several sequences acquired with different setups. First, synthetic data were created for quantitative evaluations; Second, for the evaluation on real data, two different devices were tested to recover depth maps and one or two color images were considered. The setups used in the experiments are detailed in next sections.

4.1 Synthetic Data

Synthetic data were obtained by moving a sphere in front of two moving planes. We projected this sequence in two virtual 1 Mpixel cameras. We also used one camera's depth buffer to build a depth map (see figure 2-(a)-(b)). This depth map was down-sampled to a resolution of 200×200 and used to create a meshed surface (see figure 2-(c)). We generated a translation sequence where the sphere is moving away from the cameras in front of two dynamic oblique planes, one moving upward, the other downward. It is worth noticing that extension of this framework to $N > 1$ color cameras does not change the formulation, it only stacks more constraints in equation (7).



(a) (b) (c)

Figure 2: Synthetic inputs: color image (a), depth map (b) and the computed meshed surface (c).

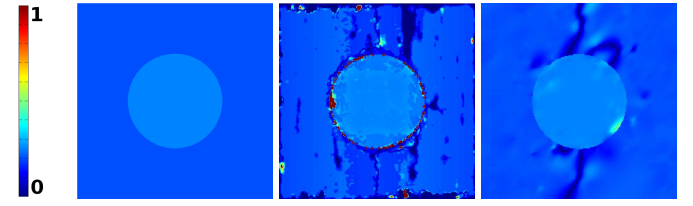
We compared our approach with the method described in [15] as the "Single camera, known scene geometry" case. This method share the same inputs as our method and is also easily extendible to the multi-camera case.

Results are presented in figure 3 where 3D motion displacement norms and directions are displayed from the camera viewpoint with a color code. Figure 4 shows the error for each vertex on the surface mesh, while table 1 presents numerical comparisons.

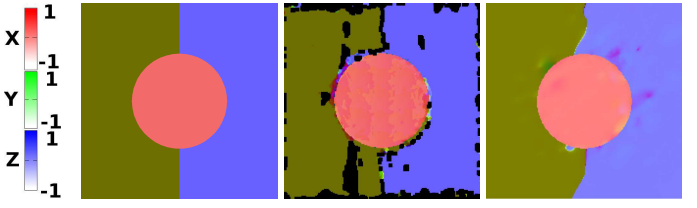
	Vedula [15]		Proposed method	
	Mean error	Median error	Mean error	Median error
Norm	33%	7.27%	8.68%	2.33%
Angle	8.6°	0.10°	2.7°	0.12°

Table 1: Numerical error on synthetic data with comparison between Vedula's method [15] and the proposed method.

As expected, results shows that the proposed method handles correctly the depth discontinuities between the sphere and the planes. Nevertheless, as there is an ambiguity on the junction between the two planes, they are considered as closely connected on the meshed surface and hence, the regularization is not performing well in this region.

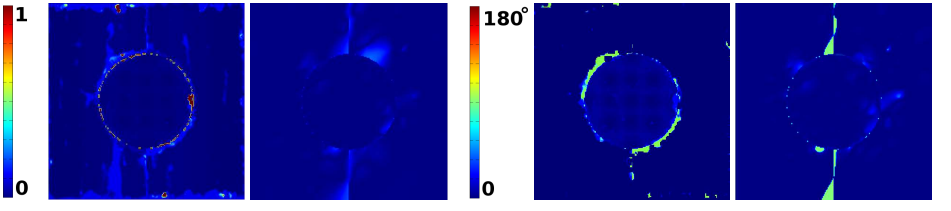


(a) 3D displacement norm coded as color



(b) 3D displacement direction coded as color

Figure 3: Results on synthetic data with comparison between ground truth (left), Vedula's method [15] (middle) and the proposed method (right)



(a) Error on 3D displacement norm

(b) Error on 3D displacement angle

Figure 4: Error on synthetic data with comparison between Vedula's method [15] (left) and the proposed method (right).

The experiments show that with highly textured synthetic data the normal flow constraint is not really improving the results as the deformation is strictly rigid and lots of feature correspondences are recovered. Adding other color cameras does not significantly improve results since recovered features from other images tend to be the same as in the first image.

4.2 Real data

We also experimented the proposed method with real data acquired with two different setups: (1) a setup composed of a Swiss Ranger SR4000 time-of-flight camera of resolution 176×144 and two 2MPixels color cameras, and (2) a calibrated Microsoft Kinect sensor, able to produce color data aligned with a depth map, both of resolution 640×480. The time-of-flight camera was calibrated along with the two color cameras using recent work presented in [5].

In order to demonstrate the ability of the proposed method to handle large motions while preserving motion discontinuities we acquired a sequence where a man is standing in a room

and playing with a ball; throwing it from one hand to the other. The sequence was repeated with all setups.

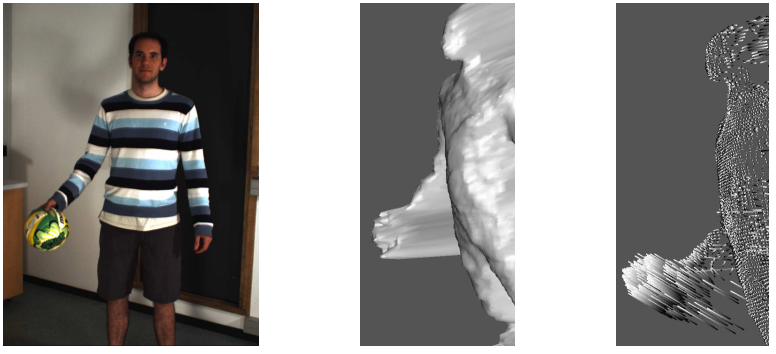


Figure 5: Input data: one of the color image (left) and the meshed surface (middle). Results: the 3D displacement field (right) on the time-of-flight data.

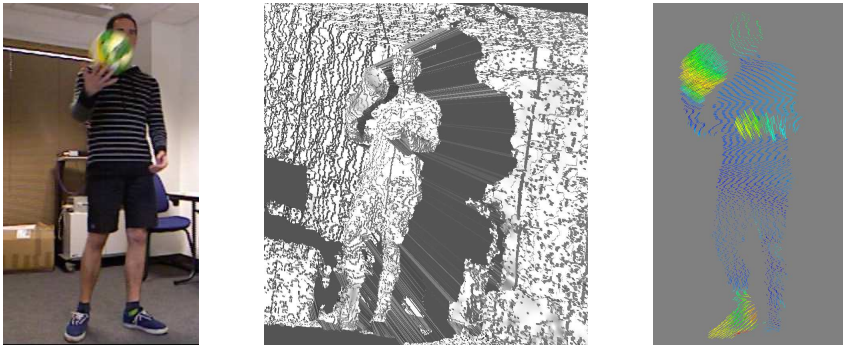


Figure 6: Input data: the color image (left) and the meshed surface (middle). Results: the 3D displacement field (right) on the Kinect data (color denotes 3D displacement norm).

Results shown on figures 5 and 6 demonstrate the interest of our method with real data taken with different setups. The recovered 3D dense motion is coherent with the action executed by the person. Using two cameras with the time-of-flight sensor helps recovering the motion even though the meshed surface is very noisy. As for the Kinect sensor, the resolution of the acquired data yields a high density mesh that increases the complexity of the linear system. In this case a parallel implementation can balance data complexity.

5 Discussion

We presented an accurate approach for 3D scene flow computation that makes use of recently available depth cameras. The main interest is to allow different kinds of photometric

information to be fused directly in 3D in an efficient way and with simple camera configurations. Using both sparse and dense image information makes this approach suitable for large displacement estimations while preserving motion details.

Short term improvements of this work include the extension to N depth cameras. This requires a prior step where depth information coming from different sensors are merged into a single surface representation. In the longer term, dense motion fields acquired this way could be used for interactive applications based on motion analysis, such as action recognition.

References

[1] J.-L. Barron, D.-J. Fleet, and S.S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 1994.

[2] T. Basha, Y. Moses, and N. Kiryati. Mutli-View Scene Flow Estimation: A View Centered Variational Approach. In *Computer Vision and Pattern Recognition*, 2010.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, 2006.

[4] M. Belkin, J. Sun, and Y. Wang. Discrete Laplace Operator on Meshed Surfaces. In *Proceedings of the Symposium on Computational Geometry*, 2008.

[5] Miles Hansard, Radu Horaud, Michel Amat, and Seungkyu Lee. Projective Alignment of Range and Parallax Data. In *Computer Vision and Pattern Recognition*, 2011.

[6] B.K.P. Horn and B.G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 1981.

[7] F. Huguet and F. Devernay. A Variational Method for Scene Flow Estimation From Stereo Sequences. In *International Conference on Computer Vision*, 2007.

[8] R. Li and S. Sclaroff. Multi-scale 3D Scene Flow from Binocular Stereo Sequences. *Computer Vision and Image Understanding*, 2008.

[9] D.G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 2004.

[10] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, 1981.

[11] J. Neumann and Y. Aloimonos. Spatio-Temporal Stereo Using Multi-Resolution Subdivision Surfaces. *International Journal of Computer Vision*, 2002.

[12] B. Petit, A. Letouzey, E. Boyer, and J.S. Franco. Surface Flow from Visual Cues. In *Vision, Modeling and Visualization Workshop, to appear*, 2011.

[13] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view Stereo Reconstruction and Scene Flow Estimation with a Global Image-based Matching Score. *International Journal of Computer Vision*, 2007.

[14] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In *European Conference on Computer Vision*, 2010.

[15] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-Dimensional Scene Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[16] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete Laplace Operators: No free lunch. In *Eurographics Symposium on Geometry Processing*, 2007.

[17] L. Xu, J. Jia, and Y Matsushita. Motion Detail Preserving Optical Flow Estimation. In *Computer Vision and Pattern Recognition*, 2010.