2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

A Primal-Dual Framework for Real-Time Dense RGB-D Scene Flow

Mariano Jaimez[1,2], Mohamed Souiai[2], Javier Gonzalez-Jimenez[1] and Daniel Cremers[2]

Abstract— **This paper presents the first method to compute dense scene flow in real-time for RGB-D cameras. It is based on a variational formulation where brightness constancy and geometric consistency are imposed. Accounting for the depth data provided by RGB-D cameras, regularization of the flow field is imposed on the 3D surface (or set of surfaces) of the observed scene instead of on the image plane, leading to more geometrically consistent results. The minimization problem is efficiently solved by a primal-dual algorithm which is implemented on a GPU, achieving a previously unseen temporal performance. Several tests have been conducted to compare our approach with a state-of-the-art work (RGB-D flow) where quantitative and qualitative results are evaluated. Moreover, an additional set of experiments have been carried out to show the applicability of our work to estimate motion in real-time. Results demonstrate the accuracy of our approach, which outperforms the RGB-D flow, and which is able to estimate heterogeneous and non-rigid motions at a high frame rate.**

I. Introduction

Estimating the motion of different objects in a scene is a topic of great relevance in robotics. From a general point of view, and without focusing on particular objects, scene flow is defined as the dense or semi-dense non-rigid motion field of a scene observed at different instants of time. Conversely to optical flow, which provides the projection of the scene motion onto the image plane, scene flow estimates the actual 3D motion field and hence requires more prior information than optical flow (2D). As a consequence, stereo or multi-view camera systems that allow for the estimation of the scene structure have been commonly employed to compute scene flow. However, the new affordable RGB-D cameras which directly provide registered RGB and depth images at a fairly high frame rate (30 Hz) are an advantageous setting for the implementation of fast scene flow algorithms.

The potential applications of scene flow in the field of robotics are numerous: autonomous navigation and manipulation in dynamic environments, pose estimation or SLAM refinement, human-robot interaction or segmentation from motion are a few examples. Nonetheless, its applicability is highly dependent on its temporal performance because the aforementioned tasks normally need to be executed at a high frame rate. Most existing approaches do not fulfill this requirement and present execution times ranging from several seconds to few hours to compute the scene flow per frame, which in practice limits their usefulness.

Research supported by the Spanish Government under project DPI1011-25483 and the Spanish grant program FPI-MICINN 2012.

[1] Department of System Engineering and Automation, University of Malaga, Spain. {marianojt, javiergonzalez}@uma.es

[2] Computer Vision Group, Department of Computer Science, Technical University of Munich {mohamed.souiai, cremers}@in.tum.de

In this paper we present the first dense real-time scene flow algorithm for RGB-D cameras. Under a variational framework, a highly parallelizable primal-dual algorithm is proposed to solve in real-time the underlying optimization problem. The benefits of this algorithm compared to direct solvers (e.g. SOR) or black box solvers are two-fold: the utilised total variation (TV) regularizer can be minimized with exactitude due to the introduction of dual variables, not needing to resort to differentiable approximations (Charbonnier penalty [1]), and a very fast version of it can be implemented on modern GPUs by reason of its first-order nature. In our variational formulation, both the optical flow and the range flow constraint equations are applied in a coarse-to-fine scheme to allow for larger displacements of points between consecutive frames. Regularization is imposed on the 3D surface in order to smooth the flow field for close points in the real space instead of for contiguous pixels in the image plane. Several experiments have been carried out to evaluate the performance of our approach and compare it with the recent work of Herbst et al. (RGB-D flow [2]). Overall, our approach achieves higher levels of accuracy in this comparison while performs 3 orders of magnitude faster. Quantitative and qualitative results show that our primal-dual scene flow is able to estimate heterogeneous and non-rigid motions precisely on a variety of scenes.

A. Related Work

Scene flow has traditionally been computed with data coming from stereo or multiple view camera systems. The term ”scene flow” was firstly coined by Vedula et al. [3] who proposed to compute the Lucas-Kanade optical flow [4] first and apply the range flow constraint equation at a later stage, obtaining a local solution which does not exploit the geometric data to estimate the optical flow. A global variational approach is presented in [5] where both the optical flow and depth flow are estimated simultaneously using quadratic regularization. To allow for discontinuities in the motion field, Total Variation (TV) was adopted in [6] and [7], where they each present a unified variational formulation to compute the disparity and the motion field jointly. However, other authors claimed that decoupling this two sub-problems is indeed advantageous and developed algorithms [8] that make the most of this decoupling to efficiently solve the global problem, combining FPGA and GPU for disparity and scene flow estimation, respectively, and leading to real-time performance. Moreover, some of the most recent works with stereo cameras [9], [10] propose to exploit the fact that in most realistic scenarios the image motion is composed of several rigid body motions, and impose local rigidity in their formulation which provides more accurate results than standard TV regularization.

The advent of RGB-D cameras few years ago represented a revolution in the field of robotics and computer vision, and much of recent research on scene flow focuses on the use of this kind of cameras. One of the first scene flow algorithms for RGB-D cameras is presented in [11], where a variational approach with quadratic data and regularization terms is proposed. Within a similar variational framework, the work of Letouzey et al. [12] makes use of a weighted quadratic regularizer and considers a set of sparse feature correspondences to handle large displacements. The regularization weights are functions of the depth gradients, and are intended to increase regularization between pixels with similar depth values and vice versa. More recently, the RGB-D flow presented in [2] achieves qualitatively good results by minimizing a functional composed of the L1-norm of the optical and range flow constraint equations and a weighted TV regularization, where the weighting function encompasses information about the depth, color and surface normals. The work presented in [13] achieves accurate results too by defining a local/global formulation with adaptive TV regularization, which is a weighted TV regularizer whose weights are similar to those presented in [12], and also resorts to interest points (SURF) to deal with large motions.

Apart from these variational approaches, there exist some Monte Carlo-based methods which provide semi-dense or dense motion fields. Cech et al. [14] presented a seed growing algorithm to compute scene flow in a stereo setup which represents a good trade-off between local and global methods with respect to accuracy and running time. Hadfield et al. [15] proposed a particle filter to estimate the motion field from depth and intensity images coming from a RGB-D camera, and extended this work in [16] to operate with any combination of photometric and range sensors. Also for RGB-D images, SphereFlow [17] exploits the availability of depth data by seeking correspondences not in the image plane but in spheres of the 3D space, which is demonstrated to be advantageous to handle occlusion and large displacements.

B. Contribution

The main contribution of this paper is a robust scene flow algorithm for RGB-D cameras that runs in real-time. To this end, a primal-dual algorithm is applied for the first time to solve the variational formulation of the scene flow problem. The particular choice of this algorithm is crucial since it is an iterative solver which performs pixel-wise updates and can be efficiently implemented on modern GPUs. Furthermore, a more natural regularization is theoretically justified and imposed, substituting the standard TV by an adaptive TV which represents the line integral of the flow field over the observed surface. Last, we take advantage of the geometric data provided by the camera to formulate a more accurate approximation of the image gradients, as well as to filter intermediate solutions in the coarse-to-fine scheme robustly.

The code is available online for public use. We also encourage the reader to watch the demonstration video at:

`http://mapir.isa.uma.es/mjaimez`

II. Variational Formulation for Scene Flow

We consider the problem of estimating the dense 3D motion field of a scene between two instants of time t and $t+1$ using color and depth images provided by an RGB-D camera. This motion field $\mathbf{M} : (\Omega \in \mathbb{R}^2) \rightarrow \mathbb{R}^3$ is defined over the image domain Ω, is described with respect to the camera reference frame and is expressed in meters per second. For simplicity's sake, an alternative representation of $\mathbf{M}$ is commonly adopted, where $\mathbf{M}$ is expressed in terms of the optical flow u, v and the range flow w. For any pixel with a nonzero depth value, the bijective relationship $\Gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ between $\mathbf{M}$ and $\mathbf{s} = (u, v, w)^T$ is given by:

$$\mathbf{M} = \Gamma(\mathbf{s}) = \begin{pmatrix} \frac{Z}{f_x} & 0 & \frac{X}{Z} \\ 0 & \frac{Z}{f_y} & \frac{Y}{Z} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{1}$$

Equation (1) can be deducted directly from the well-known "pin-hole model", where f_x, f_y are the focal length values and X, Y, Z the spatial coordinates of the observed point. Therefore, estimating the optical and range flows is equivalent to estimating the 3D motion field, but the former leads to a simplified implementation since optical and range flow constraint equations apply directly on the image plane. Hence, in order to compute the motion field we formulate a minimization problem over $\mathbf{s}$ where photometric and geometric consistency are imposed as well as a regularity of the solution:

$$\min_{\mathbf{s}} \{E_D(\mathbf{s}) + E_R(\mathbf{s})\} \tag{2}$$

In this functional, the data term E_D represents a two-fold restriction (intensity and depth matching between pairs of frames) which is insufficient to compute a unique solution. Consequently, a regularization term E_R is crucial because it does not only smooth the flow field but also further constraints the solution space.

A. Data term

Let I_0, I_1 be the intensity images and Z_0, Z_1 the depth images taken at instants t and $t+1$ respectively. We choose a data term which encourages brightness constancy and geometric consistency of the solution. The former is commonly adopted by most existing optical flow and scene flow approaches and encodes that a point should exhibit the same brightness in both intensity images:

$$\varrho_I(\mathbf{s}, x, y) = I_0(x, y) - I_1(x+u, y+v) = 0 \tag{3}$$

where x, y are the pixel coordinates in the image plane. Constancy of the intensity gradients is not considered here because our approach runs at a high frame rate, which implies that the brightness constancy assumption is quite accurate (if the source of light remains unaltered). On the other hand, depth does not remain constant over time but its change must be equal to the difference between the first depth image and the second image warped with the optical flow:

$$\varrho_Z(\mathbf{s}, x, y) = w - Z_1(x+u, y+v) + Z_0(x, y) = 0 \quad (4)$$

Conversely to many other approaches, which make use of the L_2 norm or a differentiable approximation (Charbonnier penalty) of the L_1 norm, we minimize the exact L_1 norm of (3), (4):

$$E_D(\mathbf{s}) = \int_\Omega |\varrho_I(\mathbf{s}, x, y)| + \mu(x, y)|\varrho_Z(\mathbf{s}, x, y)| dxdy \quad (5)$$

where $\mu(x, y)$ is a positive function that weights geometric consistency against brightness constancy. The L_1 norm has shown to be more robust against outliers than the L_2 norm [18] and the choice of the primal-dual algorithm to solve the minimization problem allows us to minimize it exactly without resorting to any approximation (more details will be given in section III).

The nonlinear data term E_D is nonconvex, which implies that the energy functional can have multiple local minima. In order to get as close as possible to the global minimum, we use a coarse-to-fine scheme [19] in which an image pyramid is built and the solution is computed and upsampled from coarser to finer levels, employing a linearized version of (3) and (4) at each level:

$$\varrho_I(\mathbf{s}) \approx I_0(x, y) - I_1(x+u^*, y+v^*) + \nabla I_1(x+u^*, y+v^*) \cdot (u, v)^T \quad (6)$$

$$\varrho_Z(\mathbf{s}) \approx w - Z_1(x+u^*, y+v^*) + \nabla Z_1(x+u^*, y+v^*) \cdot (u, v)^T - Z_0(x, y) \quad (7)$$

with u^*, v^* being the solution computed at the previous level. Equations (6), (7) are the well-known optical flow and range flow constraint equations whose global minimum can be obtained at each pyramid level.

B. *Regularization term*

The regularization term E_R is introduced to overcome the aperture problem associated to the optical and range flow [20], as well as to provide a smooth flow field. In this paper we present a regularizer of the flow field which is based on the total variation but takes into consideration the geometry of the scene, in contrast to standard TV which operates on the image domain Ω and disregards the real world distances between points. For the sake of clarity, a simplified 2D case shown in fig. 1 will be used to describe the proposed regularization and derive its mathematical formulation. Let $\mathbf{C} : (l \in \mathbb{R}) \to \mathbb{R}^2$ represent the observed surface of the scene (which becomes a curve in 2D) and $f : \mathbf{C} \to \mathbb{R}$ be any component of the motion field of the curve $\mathbf{C}$ with respect to the camera. The total variation of f in 2D is defined as:

$$TV(f) = \int_\Omega \left| \frac{\partial f}{\partial x} \right| dx \quad (8)$$

However, this regularization does not take into account that contiguous pixels may correspond to distant points in space with different values of f. Thus, if geometric data

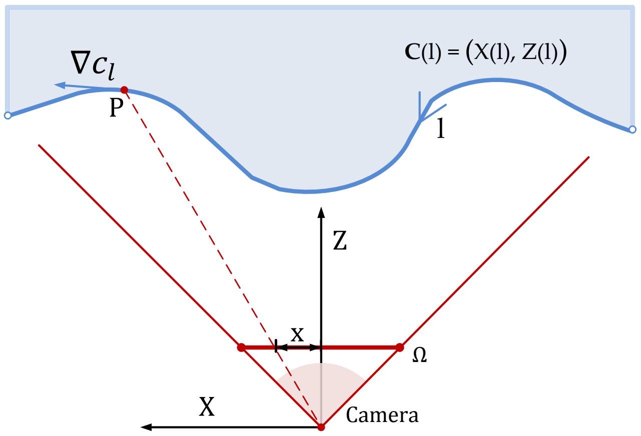

Fig. 1: Two-dimensional representation (top view) of a camera observing a scene where the image plane Ω becomes an image segment and the frontal 3D surface becomes a 2D curve $\mathbf{C}(l)$.

is available, a more natural regularization would smooth f among points which are close in $\mathbf{C}$ instead of in the image segment:

$$TV_g(f) = \int_\Omega \left| \frac{\partial f}{\partial l} \right| dx \quad (9)$$

In both cases we integrate over the image domain Ω because the curve $\mathbf{C}$ is not known, only its projection onto Ω (i.e. depth values and color information). The derivatives of f with respect to l and x can be decomposed into the two independent directions of space X, Z:

$$\frac{\partial f}{\partial l} = \frac{\partial f}{\partial X}\frac{\partial X}{\partial l} + \frac{\partial f}{\partial Z}\frac{\partial Z}{\partial l} \quad (10)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial X}\frac{\partial X}{\partial x} + \frac{\partial f}{\partial Z}\frac{\partial Z}{\partial x} \quad (11)$$

In (10), the vector $(\partial X/\partial l, \partial Z/\partial l)$ is the tangent unitary vector to the curve $\mathbf{C}$ and can be computed as a function of the gradients of the spatial coordinates X, Z over the image domain:

$$\nabla \mathbf{C}_l = \left(\frac{\partial X}{\partial l}, \frac{\partial Z}{\partial l} \right) = \frac{1}{\sqrt{\frac{\partial X}{\partial x}^2 + \frac{\partial Z}{\partial x}^2}} \left(\frac{\partial X}{\partial x}, \frac{\partial Z}{\partial x} \right) \quad (12)$$

Substituting (12) in (10) we obtain a weighted TV that takes the geometry of the scene into account:

$$TV_g(f) = \int_\Omega \left| r_x \frac{\partial f}{\partial x} \right| dx = \int_\Omega \frac{1}{\sqrt{\frac{\partial X}{\partial x}^2 + \frac{\partial Z}{\partial x}^2}} \left| \frac{\partial f}{\partial x} \right| dx \quad (13)$$

Conversely to the standard TV (8), the regularizer in (13) favors regularization between close points (high r_x) whereas it barely regularizes f for distant points (low r_x). Such behavior is very convenient since close points are likely to belong to the same object in the scene (hence moving similarly) while distant points might be part of different objects with different motion fields. Equation (13) can be easily generalized to the 3D world, leading to:

$$TV_g(f) = \int_\Omega \left| \left(r_x \frac{\partial f}{\partial x}, r_y \frac{\partial f}{\partial y} \right) \right| dxdy \quad (14)$$

$$r_x = \frac{1}{\sqrt{\frac{\partial X}{\partial x}^2 + \frac{\partial Z}{\partial x}^2}}, r_y = \frac{1}{\sqrt{\frac{\partial Y}{\partial y}^2 + \frac{\partial Z}{\partial y}^2}} \tag{15}$$

Finally, the regularization term in our variational formulation imposes the above TV penalization to the flow field:

$$E_R(\mathbf{s}) = \lambda_I \int_\Omega \left| \left(r_x \frac{\partial u}{\partial x}, r_y \frac{\partial u}{\partial y} \right) \right| + \left| \left(r_x \frac{\partial v}{\partial x}, r_y \frac{\partial v}{\partial y} \right) \right| dxdy$$
$$+ \lambda_D \int_\Omega \left| \left(r_x \frac{\partial w}{\partial x}, r_y \frac{\partial w}{\partial y} \right) \right| dxdy \tag{16}$$

Where λ_I, λ_D are constant weights which can be tuned accordingly.

III. Primal-Dual Algorithm

As pointed out in section II-A, our energy formulation is based on a linearisation of the data term. Additionally, since we utilise the convex TV regularizer in (16), this renders the overall formulation (2) convex, which makes our algorithm amenable to convex solvers. As a non-smooth energy is to be minimized, we have chosen a first order solver [21], [22] for non-smooth problems which tackles the primal-dual formulation of (2):

$$\min_{\mathbf{s}} \sup_{\substack{\boldsymbol{p_u} \in \mathcal{K} \\ \boldsymbol{p_v} \in \mathcal{K} \\ \boldsymbol{p_w} \in \mathcal{K} \\ \xi_Z \in \mathcal{Q}}} \lambda_I \int_\Omega \left\langle \boldsymbol{p_u}(x,y), \left(r_x \frac{\partial u}{\partial x}, r_y \frac{\partial u}{\partial y} \right)^\top \right\rangle dxdy +$$
$$\lambda_I \int_\Omega \left\langle \boldsymbol{p_v}(x,y), \left(r_x \frac{\partial v}{\partial x}, r_y \frac{\partial v}{\partial y} \right)^\top \right\rangle dxdy +$$
$$\lambda_D \int_\Omega \left\langle \boldsymbol{p_w}(x,y), \left(r_x \frac{\partial w}{\partial x}, r_y \frac{\partial w}{\partial y} \right)^\top \right\rangle dxdy +$$
$$\int_\Omega |\varrho_I(\mathbf{s},x,y)| \; + \; \xi_Z \; \mu(x,y) \varrho_Z(\mathbf{s},x,y) \; dxdy \tag{17}$$

where the dual variables $\boldsymbol{p_u}$, $\boldsymbol{p_v}$ and $\boldsymbol{p_w}$ corresponding to the regularizer and the dual variable ξ_Z corresponding to the data term are constrained by the following sets:

$$\mathcal{K} := \left\{ p : \Omega \to \mathbb{R}^2 \,\middle|\, |\boldsymbol{p}(x,y)|_2 \leq 1 \right\} \tag{18}$$
$$\mathcal{Q} := \left\{ \xi : \Omega \to \mathbb{R} \,\middle|\, |\xi(x,y)| \leq 1 \right\} \tag{19}$$

This iterative solver is suitable for the development of a real-time implementation of scene flow because the primal and dual pixel-wise updates can efficiently be computed in parallel on a GPU. Note that (17) is convex with respect to its primal variables and concave with respect to its dual variables which makes it possible to compute a global optimal with the primal-dual algorithm. Regarding the data term (5), we decided to just dualize the range flow $|\varrho_Z(\mathbf{s},x,y)|$ and minimize the optical flow term $|\varrho_I(\mathbf{s},x,y)|$ using the so-called "proximal" or "shrinkage" operator (see [21], [22] for further information and details about the algorithm). In the primal-dual framework, this alternative represents a good balance between fast converge and formulation complexity. On the one hand, the addition of an extra dual variable associated to the optical flow term $|\varrho_I(\mathbf{s},x,y)|$ would slow down convergence and hence the temporal performance of our method. On the other hand, formulating the primal-dual problem without dual variables associated to $|\varrho_Z(\mathbf{s},x,y)|$ and $|\varrho_I(\mathbf{s},x,y)|$ would give rise to a more involved and computationally demanding "shrinkage" operator.

IV. Implementation details

In this section we describe some important aspects associated with the implementation of the primal-dual scene flow. We will mainly focus on the computation of the image gradients, the filter applied to the solutions of each pyramid level and the weighting parameters and functions.

Regarding the image gradients, most implementations of optical or scene flow choose a particular finite difference approximation (forward, backward or centered formulas) and apply it to the whole image. Nevertheless, when this strategy is applied to compute the depth image gradients it gives rise to very high values at the edges/borders of objects, which are not representative of the real gradients of the surface (or set of surfaces) observed by the camera. Consequently, the estimated range flow w is prone to taking excessively high values at object borders owing to the inaccurate approximation of gradients adopted. As a solution, we introduce an adaptive approximation of image gradients which is consistent with the geometry of the observed scene. It consists in a weighted forward-backward formula, where the weighting functions capture the geometry derivatives in both directions (forward and backward):

$$\frac{\partial I}{\partial x} \approx \frac{r_x^+ \frac{\partial^+ I}{\partial x} + r_x^- \frac{\partial^- I}{\partial x}}{r_x^+ + r_x^-}, \; \frac{\partial Z}{\partial x} \approx \frac{r_x^+ \frac{\partial^+ Z}{\partial x} + r_x^- \frac{\partial^- Z}{\partial x}}{r_x^+ + r_x^-} \tag{20}$$

and similarly for $\partial I / \partial y$ and $\partial Z / \partial y$. The operators ∂^+ and ∂^- represent right and left derivatives and are approximated using the stencils [0 -1 1] and [-1 1 0] respectively, and the terms $r_x^+, r_x^-, r_y^+, r_y^-$ are right and left approximations of the ones presented in (15). These expressions downweight derivatives which contain borders or discontinuities (r_x and r_y very low) adopting that approximation which is more likely to capture the real surface gradient properly. If they are evaluated at pixels which do not lie on object borders then both terms will have practically the same weight and (20) will be equivalent to standard centered approximations of the image gradients.

Another important issue is the selection of an appropriate filtering strategy. It is commonly known that the solution to the variational problem might contain outliers that must be removed. These outliers are particularly detrimental in intermediate levels of the coarse-to-fine scheme since they are propagated throughout the pyramid altering significantly the final flow estimate. Traditionally, a median filter is applied to the flow estimate at each level to reject outliers. However, this filter presents an important drawback: it combines the motions of different objects when it is applied at their borders. As an alternative, we opt for using a 3×3 weighted median filter, similar to the one described in [23]. In this filter, pixels are weighted in local histograms which are accumulated to obtain the median value. The weighting function h_{median} is defined as:

$$h_{median} = \frac{1}{1 + k_d(\Delta Z)^2 + k_{dt}\frac{\partial Z}{\partial t}^2} \quad (21)$$

ΔZ measures the depth difference between pixels, $\partial Z/\partial t$ represents the temporal derivative of depth and k_d, k_{dt} are parameters. This weighting function allows us to apply a median filter that does not mix the flow fields of different objects, and at the same time penalizes pixels with high temporal derivatives, which are likely to contain outliers.

Last, further details are given about the parameters and weighting functions employed in our variational formulation. The function $\mu(x, u)$ presented in (5) which weights the geometric data term is defined as:

$$\mu(x, y) = \frac{\mu_0}{1 + k_\mu \left(\frac{\partial Z}{\partial x}^2 + \frac{\partial Z}{\partial y}^2 + \frac{\partial Z}{\partial t}^2\right)} \quad (22)$$

This definition reinforces geometric consistency in areas with low depth gradients and downweights it otherwise (high depth gradients are normally caused by jumps between objects, the presence of null depth measurements, occlusions, etc.). Furthermore, the parameters involved in the calculation of the scene flow are empirically set to the following values:

$$\lambda_I = 0.04,\ \lambda_D = 0.35,\ k_d = 5,\ k_{dt} = 10, \quad (23)$$
$$\mu_0 = 75,\ k_\mu = 1000,$$

V. Experiments

We have performed two sets of experiments to evaluate our approach quantitatively and qualitatively. Given the lack of a benchmark to test scene flow with RGB-D cameras, some authors resort to the stereo Middlebury dataset and use the depth groundtruth which, together with the intensity images from different views, emulate a pair of RGB-D images. However, this set of images does not exhibit the same characteristics that real RGB-D images: they are not affected by realistic noise or quantization effects, nor contain large empty areas (null depth) that are quite common in depth images. For these reasons, we have created a tool to test scene flow algorithms by generating artificial RGB-D images from a previously captured RGB-D frame and a predefined 3D motion field. Following this procedure, we will present quantitative and qualitative results from this semi-real data. Moreover, qualitative results for real RGB-D streams processed in real-time will be shown. In all cases the resolution of the motion field is 240×320 (QVGA).

A. Evaluation with semi-real data

In this subsection we compare our approach (named PD flow) with a recent work (RGB-D flow) [11] in terms of accuracy and temporal performance. To this purpose, we have developed a procedure to evaluate the similarity between the estimated and real motion field that exists between two RGB-D frames. This procedure is decomposed into the following steps:

1) Capture an RGB-D frame with the camera.
2) Create a 3D colored mesh from the RGB-D image.
3) Generate an artificial motion field consistent with the geometry of the scene.
4) Apply this motion field to the vertices of the mesh.
5) Generate new intensity and depth images from this deformed mesh.

As a result, a second RGB-D frame is created from a real RGB-D image and a known motion field, which can be used as a groundtruth. The resulting intensity and depth images mainly differ from real ones in one aspect: they do not contain any new information respect to the first RGB-D frame. Thus, some pixels (mainly at the image borders) might not observe the deformed mesh, and their intensity and depth values are set to zero. On the other side, the intensity values at pixels with null depth measurements are neglected (set to zero) because they are not used to build the 3D mesh (their 3D location is unknown) and they will not appear in the second frame. In any case, this last factor is not very relevant since scene flow cannot be evaluated at pixels with null depth measurements (the Γ transformation in (1) becomes singular).

Five pairs of RGB-D images have been chosen for the evaluation. These images have been generated according to the aforementioned procedure using distinct motion fields with maximum displacements ranging from 7 to 15 centimeters. Besides, these images emcompass information of realistic scenes with close and distant objects moving differently. As in similar works [16], two error measurements are compared: the 3D average angle error (AAE) expressed in degrees and the normalized root mean square error of the velocity magnitude (NRMS-V), where the maximum magnitude of the motion field (MAX-V) is used for normalization. Quantitative results are presented in table I, where two versions of the PD flow with standard TV (8) and TV$_g$ (9) along with the RGB-D flow are evaluated. On the other hand, qualitative results are displayed in fig. 2, which shows that the RGB-D pairs generated for this evaluation contain varied and heteregeneous motion fields. In fig. 2 it can be noticed that RGB-D flow is slightly more accurate than our approach computing the optical flow, whereas PD flow provides considerably better results for the range flow. Regarding the quantitative results, PD-flow with regularization on the 3D surface outperforms the other approaches. Overall, PD flow is 50% more accurate than RGB-D flow estimating the norm of the motion field and more that 100% more accurate obtaining its direction in space.

As far as temporal performance is concerned, average runtimes are measured for both methods, including different CPU - GPU implementations of our work (table II). The test platform used is a standard desktop PC running Ubuntu 14.04 with an AMD Phenom II X6 1035T CPU at 2.6 GHz, equipped with an NVIDIA GTX 780 GPU with 3GB of memory. Table II shows that even the CPU implementation of PD-flow is one order of magnitude faster than RGB-D flow (it should be remarked that RGB-D flow uses GPU). With GPU acceleration, the execution time of our primal-

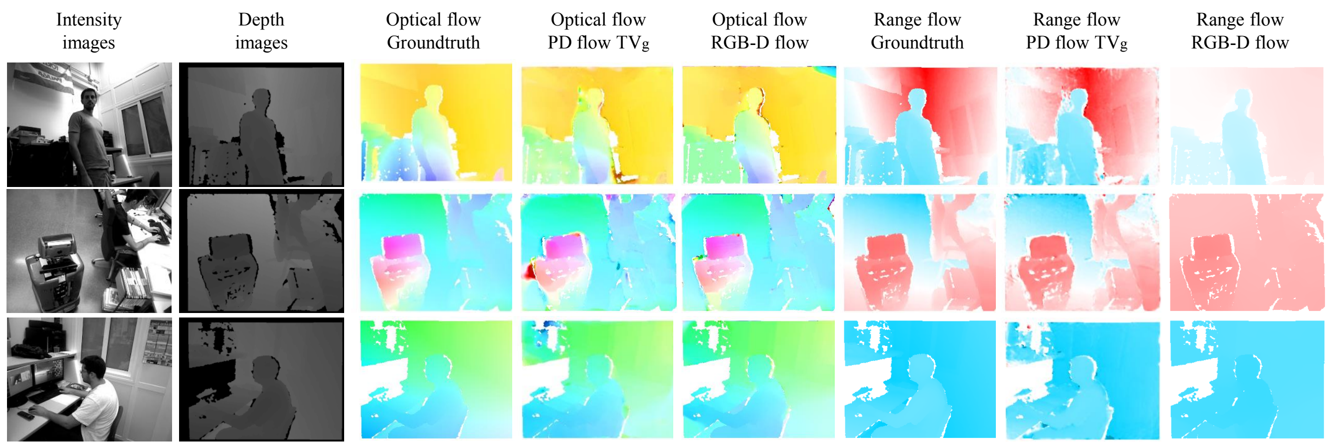

Fig. 2: Color representation of the optical and range flows for the "person standing" frames (first row), "robot top" frames (second row) and "person desk" frames (third row) following the Middlebury color scheme (for the range flow we only consider the colors of the x axis). The flow field is shown in white for the areas with null depth measurements.

TABLE I: Quantitative evaluation of scene flow with five distinct RGB-D frame pairs.

	NRMS-V PD-flow with TV$_g$	NRMS-V PD-flow with TV	NRMS-V RGB-D flow	AAE PD-flow with TV$_g$	AAE PD-flow with TV	AAE RGB-D flow	MAX-V (meters)
person standing	0.069	0.081	0.119	7.433	8.788	24.47	0.073
person desk	0.068	0.076	0.093	4.852	4.402	6.553	0.064
robot top	0.061	0.111	0.072	8.078	11.72	23.08	0.110
robot front	0.064	0.075	0.133	7.823	9.747	18.86	0.155
room	0.078	0.077	0.062	5.081	7.786	4.947	0.154
Overall	0.068	0.084	0.096	6.653	8.489	15.58	0.111

dual scene flow is 2800 times faster than the RGB-D flow, hence reaching a maximum frame rate of 24 Hz. If 30 Hz are needed, the primal-dual solver can be stopped before convergence, providing results which are only 16% less accurate than those presented in table I.

TABLE II: Runtime comparison.

RGB-D flow	PD flow CPU	PD flow GPU
119.1s	7.150s	0.042s

B. Evaluation with real data in real-time

In this subsection we show qualitative results of the scene flow working at a high frame rate (24 - 30 Hz). We have created two alternative representations of the flow field to illustrate its performance graphically. In the first case, scene flow was computed at 30 Hz (favoring speed) while two people were playing with a basketball. This experiment comprises fast movements, occlusions and heterogeneous and non-rigid motion fields associated to the different objects of the scene. A temporal sequence of pictures is shown in fig. 4, where the color goes from grey/blue for still objects (null norm of the 3D velocity) to intense red for fast motions. It can be observed that the motion field is coherent with the scene, adopting the arms/hands and the ball different red tones while the rest of the scene remains virtually still. Only the small regions at the background that are occluded by the ball in subsequent frames show a wrong (reddish) flow. It can even be noticed from the sequence that the maximum speed is reached at the moment of launching the ball, and this speed slightly decreases when the ball ascends, which is in accordance with the basic principle of energy conservation.

On the other hand, a distinct representation is displayed in fig. 3 to illustrate how the scene flow at 24 Hz (favoring precision) reproduces the real movements of a person. Two 3D point clouds are generated from consecutive depth images, being the first one shown in red and the last one in turquoise, and a vector field represents in blue the magnitude and direction of the estimated motion field. The three images in fig. 3

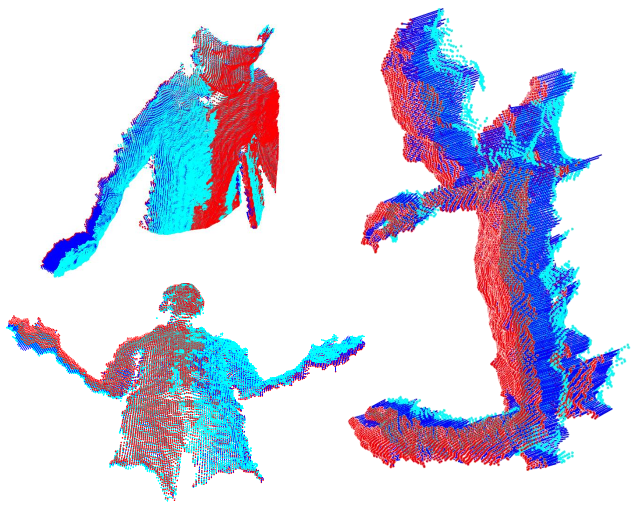

Fig. 3: 3D representation of the estimated motion field for non-rigid movements of a person. The initial and final point clouds are shown in red and turquoise respectively, and the 3D flow is depicted with blue lines.

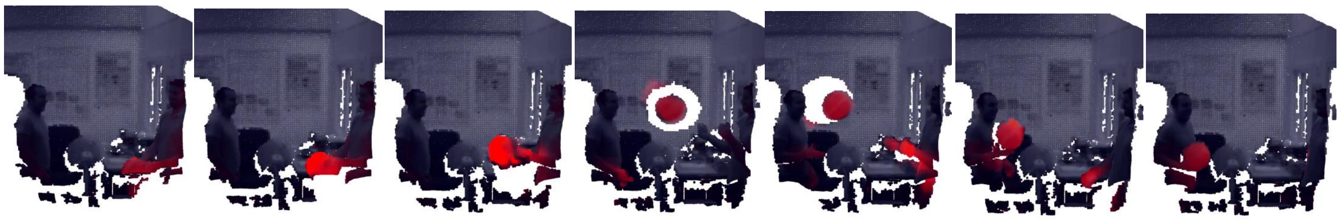

Fig. 4: Temporal sequence of two people throwing and receiving a basketball. The magnitude of the motion field is represented by colors ranging from grey/blue (null velocity) to intense red (fast motion).

show how our primal-dual scene flow is able to estimate non-homogeneous movements accurately (hand movements in the left images are significantly different than the movement of the body). In contrast to most existing approaches, we do not need to cope with very big displacements since our approach can be executed at a high frame rate, and hence the changes between consecutive images are drastically smaller for a given motion of the scene. For instance, at a frame rate of 24 Hz and considering a maximum displacement of 15 centimeters between frames (see table I) movements faster than 3 meters per second can be estimated properly.

VI. Conclusion

A novel scene flow algorithm for RGB-D cameras has been presented. Within a variational framework, geometric data from depth images is exploited to obtain more accurate results: TV regularization is applied over the observed 3D surface and a non-constant expression is proposed to approximate the image gradients regarding the geometry of the scene. In order to minimize our functional, a highly parallelizable primal-dual solver is proposed and implemented on GPU to achieve real-time performance. Our algorithm's runtime is between two and three orders of magnitude faster than previous work in scene flow for RGB-D cameras, which makes it suitable for a fair amount of robotic applications that require real-time processing. Quantitative and qualitative results are presented to demonstrate the robustness and accuracy of our approach. The code is available online under an open source license.

VII. Acknowledgements

The authors want to thank the Computer Vision group in TUM and the MAPIR group in Malaga for all the fruitful discussions and cooperation. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of hardware used for this research.

References

[1] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[2] E. Herbst, X. Ren, and D. Fox, "RGB-D flow: Dense 3-D motion estimation using color and depth," in *Int. Conf. on Robotics and Automation (ICRA)*, pp. 2276–2282, 2013.

[3] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-Dimensional scene flow," in *Proc. Int. Conference on Computer Vision (ICCV)*, vol. 2, pp. 722–729, 1999.

[4] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc.7th International Joint Conference on Artificial Intelligence*, (Vancouver), pp. 674–679, 1981.

[5] Y. Zhang and C. Kambhamettu, "On 3D scene flow and structure estimation," in *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–778, 2001.

[6] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 6–21, 2013.

[7] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proc. Int. Conference on Computer Vision (ICCV)*, pp. 1–7, 2007.

[8] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[9] C. Vogel, K. Schindler, and S. Roth, "3D scene flow estimation with a rigid motion prior," in *Proc. Int. Conference on Computer Vision (ICCV)*, pp. 1291–1298, 2011.

[10] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *iccv*, pp. 1377–1384, 2013.

[11] J.-M. Gottfried, J. Fehr, and C. S. Garbe, "Computing range flow from multi-modal Kinect data," in *Advances in Visual Computing*, pp. 758–767, Springer, 2011.

[12] A. Letouzey, B. Petit, and E. Boyer, "Scene flow from depth and color images," in *Proc. British Machine Vision Conference (BMVC)*, pp. 46.1–46.11, September 2011.

[13] J. Quiroga, F. Devernay, J. L. Crowley, *et al.*, "Local/global scene flow estimation," in *Proc. Int. Conference on Image Processing*, 2013.

[14] J. Cech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3129–3136, 2011.

[15] S. Hadfield and R. Bowden, "Kinecting the dots: Particle based scene flow from depth sensors," in *Proc. Int. Conference on Computer Vision (ICCV)*, pp. 2290–2295, 2011.

[16] S. Hadfield and R. Bowden, "Scene particles: Unregularized particle based scene flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 564–576, March 2014.

[17] M. Hornacek, A. Fitzgibbon, and C. Rother, "SphereFlow: 6 DoF scene flow from RGB-D pairs," in *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3526–3533, 2014.

[18] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for TV-L1 optical flow," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, pp. 23–45, Springer, 2009.

[19] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conference on Computer Vision (ECCV)* (T. Pajdla and V. Hlavac, eds.), vol. 3024, (Prague), pp. 25–36, Springer, 2004.

[20] H. Spies, B. Jähne, and J. L. Barron, "Range flow estimation," *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 209–231, 2002.

[21] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the piecewise smooth Mumford-Shah functional," in *Proc. Int. Conference on Computer Vision (ICCV)*, 2009.

[22] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[23] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Proc. Int. Conference on Computer Vision (ICCV)*, pp. 49–56, 2013.