Motion Cooperation: Smooth Piece-Wise Rigid Scene Flow from RGB-D Images

Mariano Jaimez[1,2] Mohamed Souiai[1] Jörg Stückler[1] Javier Gonzalez-Jimenez[2]
Daniel Cremers[1]
[1]Technische Universität München, {jaimez, souiai, stueckle, cremers}@in.tum.de
[2]Universidad de Málaga, {marianojt,javiergonzalez}@uma.es

Abstract

We propose a novel joint registration and segmentation approach to estimate scene flow from RGB-D images. Instead of assuming the scene to be composed of a number of independent rigidly-moving parts, we use non-binary labels to capture non-rigid deformations at transitions between the rigid parts of the scene. Thus, the velocity of any point can be computed as a linear combination (interpolation) of the estimated rigid motions, which provides better results than traditional sharp piecewise segmentations. Within a variational framework, the smooth segments of the scene and their corresponding rigid velocities are alternately refined until convergence. A K-means-based segmentation is employed as an initialization, and the number of regions is subsequently adapted during the optimization process to capture any arbitrary number of independently moving objects. We evaluate our approach with both synthetic and real RGB-D images that contain varied and large motions. The experiments show that our method estimates the scene flow more accurately than the most recent works in the field, and at the same time provides a meaningful segmentation of the scene based on 3D motion.

1. Introduction

Scene flow estimation has many applications such as human body pose tracking, articulated object modelling for virtual/augmented reality or traffic scene understanding. In many scenarios, the dynamic scene is composed of rigid parts: human/animal bodies, man-made articulated objects, cars in a street scene, etc. Many existing methods that work on scene flow do not completely exploit this aspect, and estimate motion fields that are only locally rigid or not rigid at all. Other methods do segment the scene to impose rigidity or strong regularization over the regions (or segments). However, these segmentations are only used as tools to improve the accuracy of the estimates, and do not really correspond to the underlying/independent motions of the scene (e.g. [20] partitions the scene into depth layers, [24] divides the scene into piecewise planar regions). Therefore, the segmentation-from-motion problem, which can be particularly useful for scene understanding or human-machine interaction, is not trully addressed by these methods.

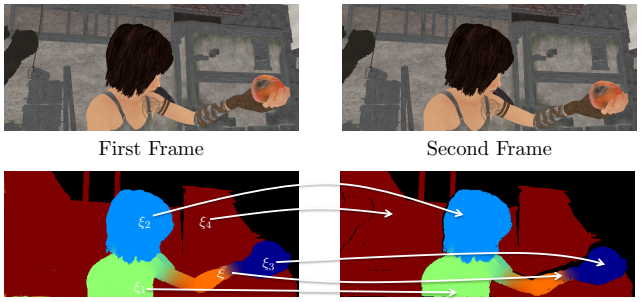


Figure 1. The proposed method is based on a motion interpolation model, which allows the emergence of smooth transitions between the segments where the motion is given by a convex combination of adjacent rigid motions (e.g. in $\bar{\xi}$).

On the other hand, assuming purely rigid motions is a strong restriction that is barely fulfilled in organic shapes. When a person moves, there are parts of their body moving rigidly (e.g. upper and lower arms or legs) and others which are transitions between the rigid ones (e.g. the neck). Besides, rigid motions within a fine-grained articulated structure may not be observable with the limited resolution of a camera. For these reasons, a sharp segmentation will never be able to estimate the motion of life beings or some other inanimate objects with exactitude.

In our method, we leverage the natural rigid-part decomposition by allowing for smooth continuous transitions between the parts. We formulate the problem of retrieving a smooth segmentation along with the motion estimates of the rigid parts, where each rigid part is assigned an independent 6 degree-of-freedom motion. To this end, we solve a non-convex optimization problem by means of coordinate descent consisting of a motion estimation step (in the fash-

ion of visual odometry) and a subsequent variational multi-labelling solver. By using a weighted quadratic regularizer over the discontinuity-preserving total variation (TV), we promote smooth transitions between motion models rather than a harsh competition. For this reason, we refer to this approach as "motion cooperation" as opposed to the traditional "motion competition". We evaluate our motion cooperation scene flow (MC-Flow) algorithm with synthetic and real RGB-D image pairs, and compare it with state-of-the-art approaches. In all cases, our approach achieves a superior performance both qualitatively and quantitatively. Furthermore, this evaluation demonstrates that the combination of a convex relaxation labelling with quadratic regularizer is superior to a sharp traditional segmentation because it naturally relaxes the overly constraining assumption of piecewise rigidity. Additionally, we show that our method retrieves meaningful soft segmentations into rigid parts as depicted in Figure 1.

1.1. Related work

Scene flow estimation has been traditionally investigated in the multi-view stereo setting within the computer vision community. Vedula *et al.* [22] have proposed one of the first methods based on the optical and range flow constraints. This approach has been later extended to regularize the flow field using quadratic [27] and TV regularization [2, 10], the latter optiziming for disparity and flow jointly. In [25], disparity and scene flow estimation has been decoupled to achieve real-time performance with a stereo camera system. The approach in [23, 24] oversegments the image into superpixels, assumes the superpixels to cover planar regions, and estimates a rigid-body motion for each superpixel individually. In [24], the planar motion of a superpixel acts as a regularization constraint on the scene flow of the individual pixels. Recently, with RGB-D cameras, the scene flow estimation topic has received further attention due to the availability of depth images at high framerate. Herbst *et al.* [8] used the L1 norm on a data term derived from the optical and range flow constraint equations and showed good qualitative results. Jaimez *et al.* [12] devised the first real-time dense scene flow for RGB-D images. A more natural TV regularization for the flow was proposed, where the regularization term minimizes the line integral of the scene flow gradients over the observed 3D surface. Quiroga *et al.* [16] overparametrize scene flow and estimate a 6-DoF rigid-body motion at each pixel. They regularize the flow field in this 6-DoF parametrization such that their model favors locally rigid motions. Hornacek *et al.* [9] also parametrize the flow-field with 6 DoF, but propose to match corresponding points within a spherical search range instead of traditional planar patch comparisons.

On the other hand, motion segmentation has also been studied in computer vision research. An early variational method for motion segmentation using optical flow constraints was proposed by Cremers and Soatto [7] in their work on motion competition. The name stems from the interpretation of the motion segments to compete for the boundaries through the best fit to their individual motion model. Several extensions to this method have been proposed, e.g. using non-parametric motions [3]. Unger *et al.* [21] explicitly model occlusions as an additional label in the multilabel optimization and impose a map uniqueness constraint to avoid ambiguous (non-bijective) data associations. All these methods are 2D and, hence, do not incorporate a 6-DoF motion model. Furthermore, they estimate a discrete segmentation.

3D-motion segmentation has only gained attention recently, mainly due to the current availability of GPUs and dense RGB-D cameras. Roussos *et al.* [17] propose a variational rigid-body motion segmentation and reconstruction method for monocular video. Zhang *et al.* [26] also pose 3D multi-body structure-from-motion in a variational framework. They require, however, a plane fitting step to make the method robust. Closely related to our method is the approach by Stueckler and Behnke [19]. They jointly estimate motion and segmentation of rigid bodies in an expectation-maximization framework in RGB-D video. Each motion segment is assigned one rigid-body motion, but the approach does not interpolate between the motions of the segments. Recently, Sun *et al.* [20] proposed a probabilistic approach which makes use of a depth-based segmentation to estimate motion between RGB-D images. They regularize the estimation process by retrieving a mean rigid-body motion in each layer. This approach also does not explicitly model smooth transitions of motions between layers, but allows for small deviations of the motion field from the layer's mean motion.

1.2. Contributions

The MC-Flow algorithm is the first approach to perform joint soft-labelling and scene flow estimation by dissecting the scene into differently moving regions and their underlying motion. Our contributions are the following:

- Our algorithm estimates 3D motion based on a smooth piecewise rigidity assumption and simultaneously finds a soft motion-based segmentation of the scene.
- By choosing a suitable regularizer we are able to interpolate between rigid motions in order to recover non-rigidly moving parts and their underlying motion.
- An arbitrary (and previously unknown) number of rigid parts can be segmented automatically.
- MC-Flow outperforms state-of-the-art RGB-D scene flow algorithms qualitatively and quantitatively.

2. Problem formulation

In this work, we assume that the scene can be segmented into n unknown distinct motion labels, each label standing for one rigid motion, as well as non-rigid parts which can be explained by neighbouring rigid motion labels. An illustration of such a smooth segmentation can be seen in Figure 1. As inputs, a pair of RGB-D frames (I_1, Z_1) and (I_2, Z_2) is given, where $I_{(.)} : \Omega \to \mathbb{R}$ and $Z_{(.)} : \Omega \to \mathbb{R}$ stand for the intensity and depth images defined on the image domain $\Omega \subset \mathbb{R}^2$. The segments and the rigid motions associated to them are obtained by minimizing a functional which depends on an implicit labelling function $u : \Omega \to [0,1]^n$, the 6-dimensional twist parametrizations $\xi_i \in \mathbb{R}^6$ of the rigid motions and the number n of rigidly moving parts. The label assigment function u encodes the moving scene in the following way:

$$u_i(x) = \begin{cases} 1 & \text{if } x \in \Omega_i, \\ 0 & \text{if } x \notin \Omega_i, \\ (0,1) & x \text{ belongs partially to } \Omega_i \end{cases} \tag{1}$$

Here we denote the i-th segment by $\Omega_i \subset \Omega$, which moves with a velocity ξ_i. Note that, in order to allow for fuzzy assignments, the label functions u_i can take on values in the interval $[0,1]$, in contrast to classical label assignment problems and their underlying binary representation.

The general problem of jointly solving for motion segmentation and motion estimation can be stated as the following optimization problem:

$$\begin{aligned} E_m(\xi, u, n) = \int_\Omega G(\xi, I_1, I_2, Z_1, Z_2, u, n)\, dx + R(u, n) \\ s.t. \sum_{i=1}^{n} u_i(x) = 1,\ u_i(x) \geq 0 \quad \forall x \in \Omega \end{aligned} \tag{2}$$

The function G encodes geometric and photometric consistency between the RGB-D images according to a linear combination of rigid-body motions:

$$\begin{aligned} G(\xi, I_1, I_2, Z_1, Z_2, u, n) = F(I_1(x) - I_2(\mathcal{W}_{\bar{\xi}}(x))) \\ + F(\left| g_{\bar{\xi}}\, \pi^{-1}(x, Z_1(x)) \right|_z - Z_2(\mathcal{W}_{\bar{\xi}}(x))) \end{aligned} \tag{3}$$

with

$$\bar{\xi} = \sum_{i=1}^{n} u_i(x)\xi_i \quad , \quad \mathcal{W}_\xi(x) = \pi(g_\xi \pi^{-1}(x, Z_1(x)))$$

and $|\bullet|_z$ meaning the z-coordinate. The warping function $\mathcal{W}_\xi(x)$ involves a projection π which transforms the 3D coordinates of the observed points into pixel coordinates. The function g relates twist coordinates to rigid transformation matrices in $SE(3)$. The function F in (3) measures photometric / geometric consistency and can be chosen according to the application and prior knowledge. In order to obtain a compact labelling, we regularize the labels by imposing a smoothing term $R(u, n)$ in (2). Note that problem (2) is hard to minimize because the labels are non-linearly involved in the non-convex dataterm G. To the best of our knowledge, except for performing complete search on u, which is unfeasible in our application, there is no direct way of tackling problem (2). Consequently, we consider a simpler formulation where the labels are pulled out of the dataterm. This significantly facilitates the optimization process because the label assignment function u is now linearly involved with the dataterm:

$$\begin{aligned} E_r(\xi, u, n) = \sum_{i=1}^{n} \int_\Omega u_i D(\xi_i, I_1, I_2, Z_1, Z_2)\, dx + R(u, n) \\ s.t. \sum_{i=1}^{n} u_i(x) = 1,\ u_i(x) \geq 0 \quad \forall x \in \Omega \end{aligned} \tag{4}$$

The data fidelity term D_i is now evaluated for every independent rigid motion:

$$\begin{aligned} D(\xi_i, I_1, I_2, Z_1, Z_2) = F(I_1(x) - I_2(\mathcal{W}_{\xi_i}(x))) \\ + F(\left| g_{\xi_i}\, \pi^{-1}(x, Z_1(x)) \right|_z - Z_2(\mathcal{W}_{\xi_i}(x))) \end{aligned} \tag{5}$$

The optimization problems (2) and (4) would be equivalent if the labels u were binary. The main difference between the two models is that in (2) the motions are interpolated and subsequently used to evaluate the residuals with the exact velocities, whereas in (4) the residuals are computed for each independent rigid motion and interpolated afterwards. With binary labels, there would not be interpolation between motions or residuals and, hence, both models would turn out to be the same. In this work, we aim to solve the motion interpolation model (2) but, given its complexity, we resort to the simpler model (4) as an approximation of (2) to optimize for the labels. For this reason, the regularization term $R(u, n)$ plays a crucial role to estimate accurate interpolated motions at the transitions between rigid bodies/parts.

2.1. Overall Optimization

Independently of which of the two models we chose, the datatermes are nonlinear with respect to the rigid motions. Therefore, the overall optimization problem is not convex and the global minimum cannot be guaranteed to be found.

To tackle this joint problem, we propose a coordinate descent strategy that alternates between estimating the motions for a fixed set of labels and then refining these labels for the recently obtained velocities, as illustrated in Algorithm 1. The motions are computed in the fashion of a visual

odometry problem, but considering that the whole scene is not rigid but smooth-piecewise rigid. The labels are solved using the approximate model (4) that is convex in u. Note that we are implicitly optimizing for the label count n by adapting the number of labels within the inner iterations, as will be described in section 5. Next, we elaborate on how to solve the main two subproblems in Algorithm 1.

Algorithm 1 Coordinate Descent Optimization for joint Motion Estimation and Segmentation

Initialize u^0

for $k = 0, 1, 2, ...$

1: $\xi^{k+1} = \arg\min_{\xi} E(\xi, u^k)$

2: $u^{k+1} = \arg\min_{u} E(\xi^{k+1}, u)$

3: Update n

end for

3. Motion estimation

Given a precomputed set of labels, at every iteration of Algorithm 1 we need to estimate the rigid-body motions associated to each label (step 1). This problem can be considered as an extension of the well-known visual odometry (VO) problem. In this more general case, the whole scene is not supposed to be moving rigidly; instead, we assume that there are n predominant rigid motions that can be linearly combined to explain the motion of every point of the scene.

Our solution to estimate the motion of the segments builds upon two existing VO methods: DIFODO [11] and the Robust Dense Visual Odometry [13]. This solution is obtained by minimizing the photometric and geometric residuals, defined as

$$r_I(x) = I_1(x) - I_2(\mathcal{W}_{\bar{\xi}}(x)) \tag{6}$$

$$r_Z(x) = \left| g_{\bar{\xi}}\, \pi^{-1}(x, Z_1(x)) \right|_z - Z_2(\mathcal{W}_{\bar{\xi}}(x)) \tag{7}$$

Note that the residuals are defined here according to the motion interpolation model (2). To cope with large motions, the process of minimization is applied in a coarse-to-fine scheme where the residuals are linearized at each level of the pyramid. In order to deal with outliers and to provide an accurate motion estimate, a robust function of the residuals is minimized:

$$\xi = \arg\min_{\xi} \left\{ \int_{\Omega} F(r_I) + \alpha F(r_Z) dx \right\} \tag{8}$$

$$F(r) = \frac{c^2}{2} ln \left(1 + \left(\frac{r}{c} \right)^2 \right) \tag{9}$$

The function F is equivalent to the Cauchy M-estimator. Although we do not present comparisons in this regard, it was chosen because it provides considerably better results than other more common choices like the L2 or L1 norms. The parameter α balances the two kinds of residuals and c controls the relative weighting between high and low residuals. This minimization problem is solved using Iteratively Reweighted Least Squares (IRLS), where the associated weighting function is

$$w(r) = \frac{1}{1 + \left(\frac{r}{c} \right)^2}. \tag{10}$$

With this strategy, we are able to solve the motion estimation problem accurately. The minimization of both the photometric and the geometric residuals allows us to estimate the motion of the segments even if they lack of texture or geometric distinctive features. This aspect is crucial because the segments can be considerably small (compared to the whole scene) and might not present sufficient photometric or geometric data to solve the 3D registration problem using only one of these two input data.

4. Label optimization

Once the motion ξ^{k+1} at a given iteration $k+1$ is obtained, we optimize the label assignment function as the second step of the overall optimization problem (Algorithm 1). For a fixed set of motions ξ, the functional $E(\xi^{k+1}, u)$ is convex and can be solved using state-of-the-art first-order solvers. In this work, the labelling function is optimized with the primal-dual algorithm developed by Pock *et al.* [15]. Detailed information about how to apply this algorithm to the addressed problem is given in the supplementary material.

In this work, two different regularizers are considered: total variation and quadratic regularization. Furthermore, the geometrical data that RGB-D cameras provide are exploited to regularize the labels according to the real 3D distances between points. Thus, regularizers are defined as a function of a weighted gradient ∇_r of the labels, whose weights (r_x) are the inverse of the 3D distances between the points:

$$\nabla_r u_i = \left(r_{x_1} \frac{\partial u_i}{\partial x_1}, r_{x_2} \frac{\partial u_i}{\partial x_2} \right) \tag{11}$$

More details on the theory and the implementation of this regularization strategy can be found in [12].

4.1. Total Variation Regularization

Total variation was made popular by the seminal work of Rudin Osher Fatemi (ROF) [18] on image denoising. The most prominent properties of the TV regularizer are allowing for jumps in the solution and being a measure of perimeter of a region if applied on its indicator function. These factors made TV widely used in general reconstruction problems like image denoising [18], image deblurring

[6] and image segmentation [5, 14]. In order to incorporate TV regularization into our approach, we simply set:

$$R(u,n) = \lambda \sum_{i=1}^{n} \int_{\Omega} \|\nabla_r u_i(x)\|_1 dx \qquad (12)$$

4.2. Quadratic Regularization

As previously mentioned, TV regularization favors sharp label boundaries. However, in our segmentation we would like to obtain a smooth interface between the labels. Hence, a suitable choice to encourage smooth label transitions is the so-called Tikhonov or quadratic regularization:

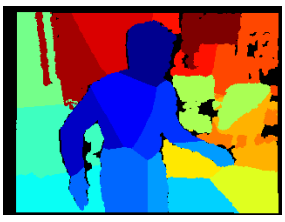
$$R(u,n) = \lambda \sum_{i=1}^{n} \int_{\Omega} \|\nabla_r u_i(x)\|_2^2 dx \qquad (13)$$

Normally, quadratic regularization does not allow for discontinuities in the solution, which would not help to provide a precise segmentation. However, the geometric weighting makes it able to estimate discontinuities in the labels and soft transitions between rigid parts at the same time.

5. Initialization and adaptive number of labels

This section describes the adopted strategy to refine the number of labels n so that they represent the actual number of independent rigid motions in the scene. Since we are solving a non-convex problem, it is crucial to start with an initial set of labels u^0 that allows us to converge to the global optimum in Algorithm 1. Instead of including the number of labels in the variational formulation (which would significantly increase the computational burden), we propose to initialize the labels with a meaningful oversegmentation of the observed scene and iteratively remove those labels that are redundant or not significant for the overall motion estimation. To this end, we create an initial K-means segmentation based on the 3D coordinates of the points of the scene. The initial number of labels is always set to 20 (the number of independent rigid motions in the scene is assumed to be smaller than this quantity). An example of a K-means initialization is shown in Figure 2. The refinement of the label count is performed after a full inner iteration of Algorithm 1 as follows:

a) Input image



b) K-means initialazation u^0

Figure 2. We initialize our algorithm by performing K-means (k=20) on the 3D coordinates of the image pixels.

- If labels i and j are associated to similar velocities, i.e., if $\|\xi_i - \xi_j\| \leq \delta$ for some small $\delta > 0$, we merge both labels.

- If a label i contains too few pixels, i.e., if $\int_{\Omega} u_i(x) < \gamma$ for some small $\gamma > 0$, we assign these pixels to the outlier label and remove label i.

6. Occlusions and outliers

In our formulation, we include an outlier label (u_n) to capture pixels with null depth measurements and those other pixels that produce very high residuals for all the possible velocity candidates ξ_i. To this end, a constant weight K_D is associated to this label which, according to (4) means that $D_n = K_D$ in the whole image plane Ω. As previously mentioned, this outlier label also plays an important role in the process of reducing the number of labels. When a label is removed as a consequence of containing very few pixels, those few pixels need to be assigned to another label. If they were assigned to a wrong label they could affect the subsequent motion estimate and spoil the results. Conversely, if they are assigned to the outlier label, they don't participate in the motion estimation stage and are automatically assigned to the best label afterwards in the label optimization stage.

On the other hand, we detect occlusions to avoid the evaluation of the dataterm (D_i in (4)) for those pixels which are not visible in the second RGB-D frame. Occlusions are handled with a binary mask $O(x)$ instead of an extra label, in a way that occluded points can still be segmented and, therefore, their 3D motion is estimated too. This can be accomplished by virtue of the regularization term, and allows us to provide a complete segmentation of the scene even if some points or areas are occluded after the motion.

In order to detect occlusions, two factors are considered: the amount of pixels that are registered to each pixel of the second frame and the temporal change in the depth images. First, we compute a cumulative function $C(x) : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}$ that counts how many pixels from the first frame are warped to the pixel x of the second frame (according to the estimated motion). Without occlusions, this function is approximately equal to 1 (or maybe inferior to one for new points appearing in the second frame), meaning that there is a one-to-one (bijective) correspondence between the observed points at both images. On the contrary, if $C(x)$ is noticeably higher than one, there are some pixels in the first frame that are warped to the same pixel x in the second frame, indicating the existence of occlusions. Consequently, we can define a function $O_C(x)$ that finds the pixels candidates for occlusion by applying a warping with the estimated motion and evaluating the cumulative function C:

$$O_C(x) = C(\mathcal{W}_{\bar{\xi}}(x)) \qquad (14)$$

On the other hand, unlike in the optical flow problem, geometric information is available and can be exploited to reason whether a point is occluded or not. The simplest function that can be used to detect occlusions is the temporal change in depth:

$$O_Z(x) = Z_1(x) - Z_2(x) \tag{15}$$

Combining these two functions we can detect most of the occluded areas in the scene by imposing a threshold K_o:

$$O(x) = \begin{cases} 1 & \text{if} \quad O_C(x) + K_z O_Z(x) > K_o \\ 0 & \text{else} \end{cases} \tag{16}$$

where K_z is a parameter that weights O_Z against O_C. This strategy could be improved by embedding these functions into a variational formulation and imposing regularization over the occlusion mask. However, this has not been implemented in our work because it would significantly increase the runtime of our method.

7. Experiments

In this section, qualitative and quantitative results are presented to evaluate the accuracy of our approach. These results are divided into two categories: scene segmentation and scene flow estimation. However, the evaluation process is not straightforward given the lack of benchmarks with either scene flow ground truth or segmentation from 3D motion. For this reason, we have selected a set of synthetic and real RGB-D frame pairs that contain varied and challenging motions. First, our approach is tested with some sequences from the Sintel dataset [4]. This dataset contains scenes with heterogeneous and large motions, and provides optical flow ground truth which can be used to measure the scene flow error. Second, the joint segmentation and motion estimation is generated for several RGB-D image pairs that either have been utilized in previous works in the literature (as in [16]) or have been taken with RGB-D cameras in our lab. In all cases, two versions of our method are tested, corresponding to the two different regularization strategies for the label optimization problem: total variation (TV) and quadratic regularization (Quad). The resolution adopted for the images is QVGA (240×320) for those taken with an RGB-D camera and 218×512 for the Sintel sequences. The maximum depth is set to 5 meters in all cases. Tests have been performed with a total of fourteen image pairs: eight from the Sintel dataset (named "Sintel-1...8") and six real image pairs (named "RI-1...6").

7.1. Scene segmentation

In this subsection we present the motion segmentation that our method provides for all the tested sequences. The occlusion layer is also displayed for some sequences together with the segmentation although the occlusion is not a label itself (but a mask). Figure 3 shows the results for the Sintel images. It can be observed that TV produces very sharp labels with very few pixels interpolating between different motions. On the contrary, quadratic regularization gives rise to a smooth segmentation where many pixels adopt an interpolated velocity between two (or maybe more) rigid-body motions. The same behavior can be seen in Figure 4 where the results for the real RGB-D images are presented. In general, it can be noticed that the number of labels to which the method converges is not the same for the two regularization strategies. Normally, TV produces a higher number of labels because it is not able to interpolate motions and tends to keep extra labels to compensate for it. It can be observed that, but for Sintel-4 (with TV) and RI-5, the resulting segmentations represent quite accurately the different objects and rigid parts of the scenes.

7.2. Scene flow evaluation

For all the sequences, the scene flow is evaluated quantitatively and compared with three state-of-the-art methods: the Primal-Dual flow (PD-Flow) [12], the Semi-Rigid flow (SR-Flow) [16] and the Layered flow [20]. First, the photometric and geometric residuals are computed by warping the intensity and depth images (respectively) according to the estimated flow. It is important to note that occluded pixels will show very high residuals even if the motion is accurately estimated for them, which considerably disturbs the error metrics (RMSE of the residuals). To overcome this limitation and to provide more precise comparisons, we compute the RMSE of the non-occluded pixels, which is a more reliable metric of the scene flow accuracy. To this end, we assume that the occlusion layer computed by our approach is sufficiently accurate and use it in all cases (neither PD-flow nor SR-flow detect occlusions). This does not represent any bias toward our method because it is a common mask applied to all of them, and if some occluded pixels have not been detected properly then they will affect the error metrics of all the compared methods equally. Table 1 shows the results for all the frame pairs. It can be observed that our method provides the most accurate estimates with both TV and quadratic regularization. The differences between TV and Quad are essentially caused by the way they produce transitions between the labels and the number of labels they converge to. As previously analyzed, TV generates a sharp segmentation where the motion is barely interpolated, whereas quadratic regularization provides smooth transitions between the labels that lead to larger areas with interpolated motions. On the other hand, TV tends to converge to a higher number of labels, which helps to compensate for its inability to capture nonrigid motions. Overall, the best results are obtained with quadratic regularization, although the differences are small.

For the sake of clarity, Figure 5 is included to illustrate

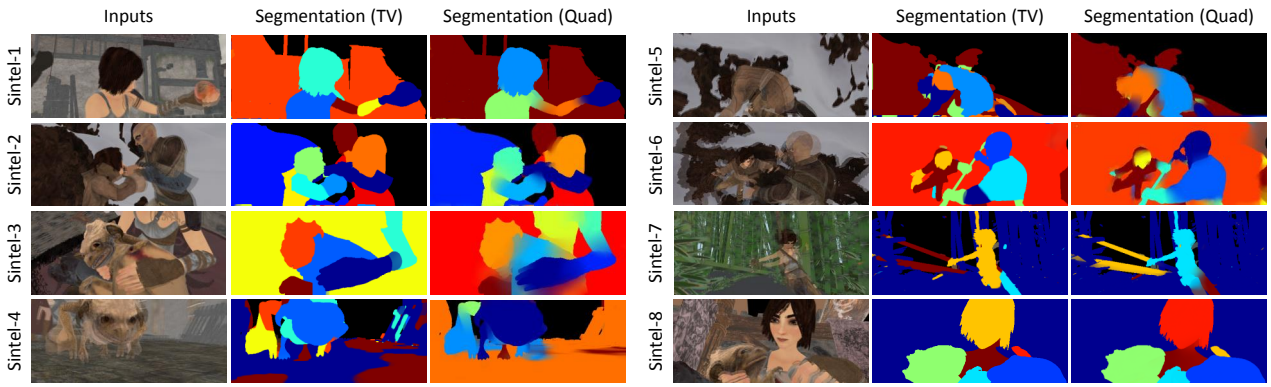


Figure 3. Segmentation estimated by our approach for the eight sequences of the Sintel dataset considered. Colors are independent for each result and do not depend on the associated rigid motion. Black represents the outlier label.

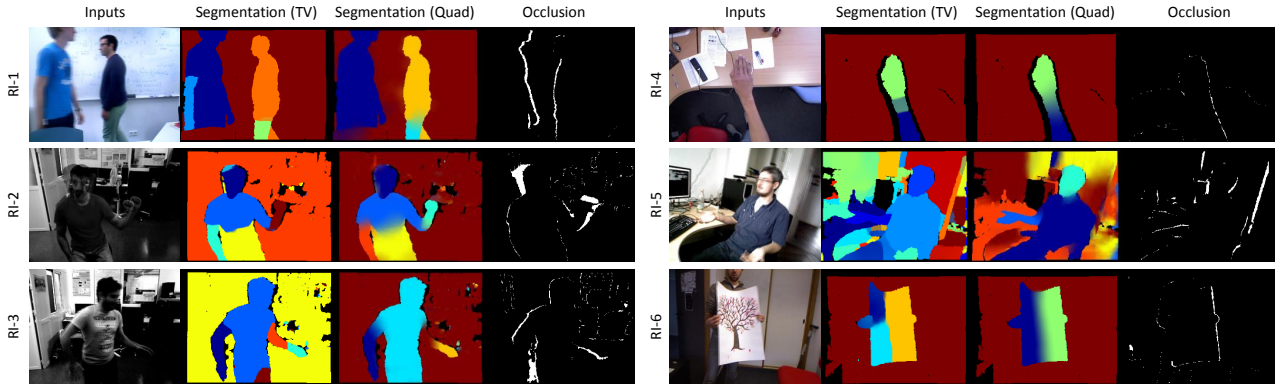


Figure 4. Segmentation and the occlusion layer estimated by our approach for 6 image pairs taken with RGB-D cameras. Colors are independent for each result and do not depend on the associated rigid motion. Black represents the outlier label.

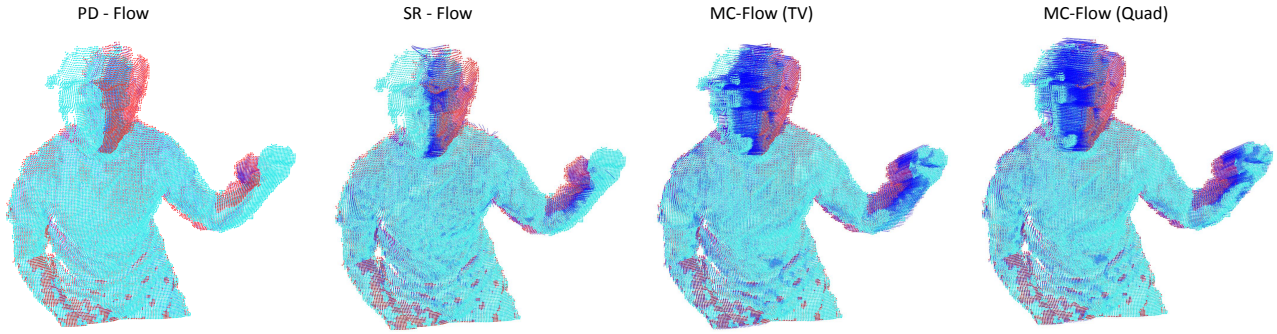


Figure 5. Comparison of the 3D motion fields estimated for the "RI-2" sequence. The initial frame is represented by the red point cloud, the final frame by the turquoise point cloud and the scene flow by the blue lines. The above comparison shows that our approach provides the most accurate estimate of the scene flow.

the 3D motion field that the compared methods estimate for the sequence "RI-2". PD-Flow, which was conceived to work in real-time, is unable to estimate large motions and can only capture the motion of the body and the upper arms. SR-Flow provides better results but is still unable to reproduce the real motion of the hand and head. Only our approach estimates the whole motion field properly, specially with quadratic regularization of the labels.

Moreover, for the Sintel image pairs, we project the scene flow onto the image plane to obtain the optical flow and compare it with the ground truth provided by the Sintel dataset. In this case we evaluate two error metrics: the average end-point error (EPE) and the average angular error (AAE), as explained in [1]. Again, the results (Table 2) are computed for the non-occluded pixels, which is a fairer comparison given that some methods do not manage occlusions and hence provide bad estimates for the occluded areas. It can be seen that our approach with both TV

	Photometric residual - RMSE					Geometric residual - RMSE				
	PD-Flow	SR-Flow	Layered-Flow	MC-TV	MC-Quad	PD-Flow	SR-Flow	Layered-Flow	MC-TV	MC-Quad
Sintel-1	0.060	0.035	0.049	0.022	**0.021**	0.443	0.317	0.420	0.253	**0.186**
Sintel-2	0.057	0.068	0.063	0.026	**0.025**	0.086	0.090	0.108	0.056	**0.053**
Sintel-3	0.048	0.041	0.047	0.032	**0.028**	0.021	0.022	0.035	0.018	**0.017**
Sintel-4	0.091	0.069	0.109	0.063	**0.044**	0.378	0.347	0.607	**0.155**	0.190
Sintel-5	0.074	0.067	0.091	**0.051**	0.055	0.373	0.267	0.498	**0.203**	0.283
Sintel-6	0.120	0.118	0.127	**0.055**	**0.055**	0.224	0.190	0.253	0.114	**0.096**
Sintel-7	0.076	0.071	0.079	**0.035**	0.038	0.407	0.423	0.382	0.233	**0.188**
Sintel-8	0.063	**0.026**	0.045	0.028	0.027	0.083	0.069	0.086	0.038	**0.037**
RI-1	0.038	0.025	0.031	0.024	**0.022**	0.070	0.060	0.046	**0.038**	**0.038**
RI-2	0.032	0.028	0.035	0.021	**0.020**	0.286	0.259	0.294	0.114	**0.102**
RI-3	0.031	0.024	0.027	**0.018**	**0.018**	0.221	0.208	0.217	0.160	**0.145**
RI-4	0.015	0.012	0.011	**0.008**	**0.008**	0.025	**0.024**	0.025	0.025	0.025
RI-5	0.074	0.051	0.056	**0.039**	0.040	0.095	0.087	0.108	**0.079**	0.085
RI-6	0.077	0.050	0.070	0.049	**0.047**	**0.036**	0.038	0.037	0.041	0.040
Average	0.061	0.049	0.060	0.034	**0.032**	0.197	0.172	0.223	0.109	**0.106**

Table 1. Photometric and geometric residuals after warping the image pairs with the estimated scene flow.

	Optical flow - EPE					Optical flow - AAE				
	PD-Flow	SR-Flow	Layered-Flow	MC-TV	MC-Quad	PD-Flow	SR-Flow	Layered-Flow	MC-TV	MC-Quad
Sintel-1	1.940	0.684	1.320	0.221	**0.219**	27.87	7.694	13.26	**2.486**	2.827
Sintel-2	2.299	2.100	2.851	0.367	**0.324**	23.63	16.02	35.50	**4.826**	4.950
Sintel-3	1.223	1.130	0.975	0.383	**0.344**	31.69	20.21	20.80	8.364	**7.721**
Sintel-4	17.04	21.68	15.26	10.23	**3.436**	73.57	90.56	43.09	22.13	**9.694**
Sintel-5	4.381	3.990	3.212	2.316	**1.983**	24.27	26.14	10.43	14.56	**10.16**
Sintel-6	6.045	7.739	7.67	**1.168**	1.498	12.10	18.99	27.52	**3.845**	5.194
Sintel-7	2.875	3.335	3.382	**1.480**	1.591	26.50	21.26	22.48	**7.723**	8.169
Sintel-8	1.674	0.456	1.012	**0.228**	**0.228**	22.45	4.713	8.003	3.762	**3.757**
Average	4.685	5.142	4.461	2.049	**1.203**	30.26	25.70	22.63	8.462	**6.559**

Table 2. Average end-point and angular errors of the optical flow computed by projecting the estimated scene flow onto the image plane.

and quadratic regularization clearly outperforms the others, providing a motion estimate that is between 2 and 5 times more accurate than those from the PD-Flow, SR-Flow and the Layered-Flow.

Regarding the computational performance, our method ranks second with a runtime of 30 seconds. For the experiments, we have utilized 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. The measured runtimes are:

- PD-Flow: 0.042 seconds (GPU).
- SR-Flow: 150 seconds (CPU).
- Layered-Flow: 8 minutes (CPU).
- MC-Flow: 30 seconds (label optimization on GPU and all the remaining steps on CPU).

8. Conclusion

In this paper we have addressed the problem of joint segmentation and scene flow estimation from RGB-D images. The overall optimization problem is solved by means of a coordinate descent method which alternates between motion estimation and label optimization, while at the same time adapts the number of labels to the real number of independent rigid motions of the scene. Two different regularization strategies for the labels are employed, TV and quadratic, leading to sharp and smooth segmentations, respectively. Our method has been tested with both synthetic and real RGB-D image pairs, and the experiments show that joint segmentation and motion estimation provides very accurate results that outperform state-of-the-art scene flow algorithms on RGB-D frames. Comparisons between the two regularization strategies show that quadratic regularization estimates motion more accurately than TV because

it generates smooth label transitions between rigid bodies, which models the scene motion more realistically. For future work, we plan to extend this work to RGB-D video streams where temporal regularization can be imposed.

9. Acknowledgement

This research was supported by the Spanish Government under the grant programs FPI-MICINN 2012 and DPI2014-55826-R (co-founded by the European Regional Development Fund), as well as by the EU ERC grant Convex Vision (grant agreement no. 240168). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of hardware used for this research.

References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 7

[2] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. *International Journal of Computer Vision*, 101(1):6–21, 2013. 2

[3] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In *Proc. European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 471–483. 2006. 2

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conference on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625, 2012. 6

[5] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences*, 5(4):1113–1158, 2012. 5

[6] T. F. Chan and J. Shen. Variational image deblurring - a window into mathematical image processing. *Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore*, 2004. 5

[7] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62:249–265, 2005. 2

[8] E. Herbst, X. Ren, and D. Fox. RGB-D flow: Dense 3-D motion estimation using color and depth. In *Proc. Int. Conference on Robotics and Automation (ICRA)*, pages 2276–2282, 2013. 2

[9] 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)*, pages 3526 – 3533, 2014. 2

[10] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 1–7, 2007. 2

[11] M. Jaimez and J. Gonzalez-Jimenez. Fast visual odometry for 3-D range sensors. *IEEE Transactions on Robotics*, 31(4):809–822, 2015. 4

[12] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense RGB-D scene flow. In *Proc. Int. Conference on Robotics and Automation (ICRA)*, pages 98 – 104, 2015. 2, 4, 6

[13] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *Proc. Int. Conference on Robotics and Automation (ICRA)*, May 2013. 4

[14] M. Nikolova, S. Esedoglu, and T. F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006. 5

[15] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proc. Int. Conference on Computer Vision (ICCV)*, 2011. 4

[16] J. Quiroga, T. Brox, F. Devernay, and J. Crowley. Dense semi-rigid scene flow estimation from RGBD images. In *Proc. European Conference on Computer Vision (ECCV)*, pages 567–582, 2014. 2, 6

[17] A. Roussos, C. Russell, R. Garg, and L. de Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *IEEE Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 31 – 40, 2012. 2

[18] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. 4

[19] J. Stückler and S. Behnke. Efficient dense rigid-body motion segmentation and estimation in RGB-D video. *International Journal of Computer Vision*, 113(3):233–245, 2015. 2

[20] D. Sun, E. B. Sudderth, and H. Pfister. Layered RGBD scene flow estimation. In *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 6

[21] M. Unger, M. Werlberger, T. Pock, and H. Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[22] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. Int. Conference on Computer Vision (ICCV)*, volume 2, pages 722–729, 1999. 2

[23] C. Vogel, K. Schindler, and S. Roth. 3D scene flow estimation with a rigid motion prior. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 1291–1298, 2011. 2

[24] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 1377–1384, 2013. 1, 2

[25] 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*, 95(1):29–51, 2011. 2

[26] G. Zhang, J. Jia, and H. Bao. Simultaneous multi-body stereo and segmentation. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 826 – 833, 2011. 2

[27] Y. Zhang and C. Kambhamettu. On 3D scene flow and structure estimation. In *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 778–785, 2001. 2