GraphFlow – 6D Large Displacement Scene Flow via Graph Matching

Hassan Abu Alhaija[1,2], Anita Sellent[1,3], Daniel Kondermann[2], Carsten Rother[1]

[1] TU Dresden, Dresden, Germany
hassan.abu_alhaija@tu-dresden.de
[2] Heidelberg University, Heidelberg, Germany
[3] TU Darmstadt, Darmstadt, Germany

Abstract. We present an approach for computing dense scene flow from two large displacement RGB-D images. When dealing with large displacements the crucial step is to estimate the overall motion correctly. While state-of-the-art approaches focus on RGB information to establish guiding correspondences, we explore the power of depth edges. To achieve this, we present a new graph matching technique that brings sparse depth edges into correspondence. An additional contribution is the formulation of a continuous-label energy which is used to densify the sparse graph matching output. We present results on challenging Kinect images, for which we outperform state-of-the-art techniques.

1 Introduction

In this work, we tackle a fundamental problem in computer vision – that is to estimate a dense correspondence field between a pair of images. While for some scenarios, e.g., small motion in a highly textured scene, this task is considered to be solved, there are still a number of outstanding challenges in the general case. The particular challenge we are addressing in this work is when the scene and/or the camera are subject to large movements. This occurs frequently when objects move at high speed or humans perform articulated actions, such as gesturing, walking or doing sport. Also, large displacements occur in time lapse photography, e.g., when a camera surveils a building site or observes the growth of a plant. Another scenario is when intermediate frames in a video sequence have to be deleted and the task is to find a smooth transition between the remaining frames. Unfortunately, large displacements violate the assumptions of most state-of-the-art scene flow estimation techniques, i.e. variational approaches [14, 17]. They achieve very high accuracy when the overall motion is small. However, for large displacements the main problem is to estimate the general motion of all objects correctly. The task of this work is to show how this general motion can be recovered reliably. We here ask the specific question of computing dense 6D scene flow between a pair of RGB-D images with independently moving and deforming objects. This means that for each pixel we aim at recovering the 3D translation and 3D rotation, that matches a pixel (and its local neighborhood) to the corresponding point in the other image. To compute dense flow for large

displacements, many works have proposed to first find some sparse matches between the two frames and then, subsequently, utilize this information to estimate a dense flow field, e.g. [6, 15, 29]. Distinctive points can be matched using, e.g., SURF [3] or SIFT [19]. However, this assumes that the scene contains sufficiently textured surfaces and non-repetitive patterns. While man-made environments often violate those assumptions, they are highly suitable for using active depth cameras, e.g., active stereo or time-of-flight sensors [11]. These devices provide depth maps even for untextured surfaces. In our two stage scene flow approach, instead of using sparse texture matches only, we utilize depth edges extracted from the RGB-D images that describe object boundaries well. However, in the presence of large motion, they are actually not trivially described and matched. While exact edge description suffers from occlusion and distortion effects, more robust edge descriptors often lead to ambiguous matches. To disambiguate edge matches with robust descriptors, we use a structured matching approach in the form of graph matching that profits from non-local information to assign edge matches. The structure-preserving properties of an underlying loopy graph allows the strong and unique matches to guide the weak and ambiguous ones. While building the structure graph for a moving camera in a static scene is a relatively straight forward task, in this work we show how the depth information can be used to construct 3D graphs that respect independently moving objects in addition to camera motion. In the second stage, we show how dense scene flow can be obtained from graph matching by extending the recent SphereFlow method of Hornacek et al. [15]. For this we propose a new energy function that incorporates a left-right consistency check as well as standard smoothness and data terms. The energy is optimized with alpha expansion, and we demonstrate an improvement in performance with respect to SphereFlow. To summarize, the main contributions are

- A state-of-the-art method for large displacement scene flow from RGB-D image pairs.
- A new graph matching technique that exploits depth information.
- A new continuous-label energy for scene flow that jointly models a left-right consistency check, as well as spatial smoothness and local appearance.

2 Related Work

Since the introduction of scene flow by Vedula et al. [26] numerous approaches to scene flow estimation have been proposed. Many of them use multi-view video frames as input and employ the input images to compute depth structure and 3D motion, e.g. [16, 21, 28]. However, with recent depth cameras, RGB-D images have become readily available. Variational approaches to scene flow estimation from RGB-D images, e.g. [14, 17], combine pixel-wise brightness and gradient constancy with depth velocity constraints, and additional regularization of the 3D motion, to obtain a global solution. Since they rely on iterative linearization or second order approximation, e.g. [9], variational approaches without appropriate initialization are restricted to small or moderately large motion even when

iterative warping and coarse-to-fine schemes are used. In contrast, discrete scene flow approaches demonstrate good performance also for large motion. Hadfield and Bowden [12] estimate scene flow with a particle based formulation, and can deal also with large 3D motion. However, they assume constant velocity in a multi-frame image sequence. Hornacek et al. [15] and Wang et al. [29] use a PatchMatch based algorithm [2] with a local data term to generate 6D motion proposals between two frames. For large displacement motion, these approaches define currently the state-of-the-art. But they still fail in the absence of sufficiently textured surfaces, as we will show. In our approach we extend the model of [15] by a term for the left-right consistency check, which has been done before for, e.g., variational optical flow [1].

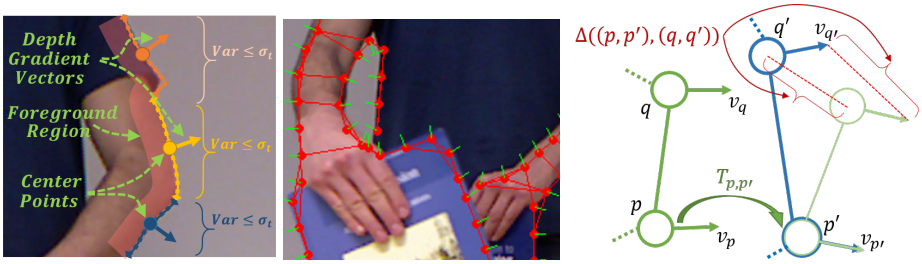
The information of landmark matching is currently exploited in a few scene flow approaches. Hornacek et al. [15] use sparse SURF feature matches in addition to random initialization; Quiroga et al. [20] match SURF features on each level of the image pyramid and encourage dense scene flow to behave accordingly. Similar approaches are known from the optical flow literature. For example, Brox and Malik [6] or Weinzaepfel et al. [30] use the strength of feature matches to support large displacement estimation while still using an image pyramid. Leordeanu et al. [18] and Revaud et al. [22] use semi-dense matches to replace the image pyramid. However, single feature matches are often too unreliable to be directly included into dense motion estimation. Sellent et al. [23] use additional images to improve feature matches, while Xu et al. [31] decide on each pyramid level anew, if and what feature matches are utilized. In our approach we use a graph matching strategy to improve the reliability of feature matches. Graph matching has a very wide field of applications in pattern recognition and machine vision, see [8, 10]. It provides non-local information on landmarks by embedding them in a graph structure. In our approach we formulate the graph structure on depth edge features. In contrast to edge or line matching in RGB images [4, 27] we can extract these features robustly. Additionally, we can use the depth channel to build the structure of the graph by avoiding to connect features across depth discontinuities. This is an advantage over, e.g., Zhang et al. [33] which cannot profit from depth information for graph construction.

3 Method

Our graph matching scene flow approach proceeds in two steps. In the first step, we determine depth edges in the two input RGB-D frames, construct the associated graphs, and then match them. In the second step, we use the sparse motion information obtained from the graph matching to assign dense, smooth and consistent 6D rigid body motion to each observed pixel in both frames.

3.1 Graph matching

For $\Omega, \Omega' \subset \mathbb{R}^2$ let $I : \Omega \to \mathbb{R}^4$, $I' : \Omega' \to \mathbb{R}^4$ be two RGB-D images with depth and color channel I_d, I_c and 3D-to-2D mapping $\pi : \mathbb{R}^3 \to \Omega$. We pre-process the



(a) (b) (c)

Fig. 1: **Details of graph matching.** (a) Our edge description segments are represented by their center point, average appearance descriptor that describe the foreground region and normalized depth gradient vector. In order to compute a description segment we accumulate neighboring edge pixels whose descriptor variance is lower than a threshold σ_t and whose count is between $r_{min} = 20$ and $r_{max} = 30$ pixels. (b) For graph matching, the description segments centers are connected to form a graph. In particular, each description segment center is connected to its $N = 3$ nearest neighbors with respect to the geodesic distance of the depth map to avoid connections across large depth changes. (c) Illustration of the geometry term $\Delta((p, p'), (q, q'))$ for graph matching, see Eq. (3) and text.

depth channel to fill-in the unknown depths via morphological operations and apply a median filter to suppress noise. We extract edges in the depth map with the Canny edge detector [7]. For each edge pixel we use the orientation of the depth gradient to determine the foreground region around this point. We use the SIFT descriptor with three different sizes $(8, 16, 32)$ [19] on the foreground region to describe the appearance of the edge point. As edges might change length and appearance between frames we do not use this pixel-wise description directly, but instead group edge pixels with similar appearance into *description segments*, see derivation in Fig. 1(a). Each description segment is represented by its center point which is the median of all its points, its normalized depth gradient vector and the mean of all its pixels' descriptors.

Based on these description segments we construct the graph structure. Let R and R' be the set of all descriptor segments centers in image I and I' respectively. For each element in R we create graph edges to its $N = 3$ nearest description segments, considering the geodesic distance of the depth map, see Fig. 1(b). Using the geodesic distance for graph edge construction we ensure that segments are not connected over depth discontinuities, which often coincide with object boundaries, hence motion boundaries. The set of all graph edges between description segments centers are represented by the graph edge sets E, E'. This gives the two graphs $G = (R, E)$ and $G' = (R', E')$ defined on image I and I' respectively. We denote by $A \subseteq R \times R'$ the set of all potential assignments between the two sets of description segment centers. A matching configuration between the two graphs is represented by the binary vector $\mathbf{x} \in \{0, 1\}^{|A|}$ where

for each $a = (p, p') \in A$ the entry $x_a = 1$ means that p matches p'. Thereby each matching configuration must satisfy a uniqueness constraint where each description segment has at most one match. Our matching objective function is

$$E(x) = \lambda^{app} E^{app}(x) + \lambda^{geom} E^{geom}(x) + \lambda^{occ} E^{occ}(x) \ . \tag{1}$$

The energy consists of three terms, each weighted individually (here we use $\lambda^{app} = 300, \lambda^{geom} = 0.1, \lambda^{occ} = 50$).

The **appearance term** E^{app} is a unary term that measures similarity in appearance of matched descriptor segments. We use the Euclidean distance between the concatenated SIFT feature vectors and set

$$E^{app}(x) = \sum_{a=(p,p') \in A} ||desc(p) - desc(p')||_2 \ x_a \tag{2}$$

where $desc(p)$ is the average descriptor of the description segment of center p.

The **geometry term** E^{geom} is a pairwise term that defines the relationship between pairs of neighboring assignments, see Fig. 1(c). The tuples $a = (p, p')$ and $b = (q, q')$ are in a neighbor set N when either the edge $\overrightarrow{pq} \in E$ or $\overrightarrow{p'q'} \in E'$ exist. Let $T_{p,p'}$ be the 2D translation and rotation that maps $\overrightarrow{pv_p}$ to $\overrightarrow{p'v_{p'}}$, where $v_p = p + d_p$, $v_{p'} = p' + d_{p'}$ and $d_p, d_{p'}$ are the normalized depth gradient vectors at p, p' respectively. A geometry preserving matching should then map neighboring description segments to description segments that satisfy a similar transformation, i.e. $\Delta((p, p'), (q, q')) = ||T_{p,p'}(q) - q'||_2 + ||T_{p,p'}(v_q) - v_{q'}||_2$ should be small. Thus our geometry term

$$E_{geom}(x) = \sum_{(a,b) \in N} (\Delta(a, b) + \Delta(b, a)) \ x_a \ x_b \tag{3}$$

penalizes differences in length between the vectors $\overrightarrow{pq}$ and $\overrightarrow{p'q'}$ and inconsistency in their rotations. Meanwhile, arbitrary consistent rotations are allowed. Note that in [25] the pairwise term penalizes all rotations regardless of their consistency with neighbors, which had in our experiments a negative effect.

The **occlusion term** E^{occ} penalizes unmatched description segments by adding a negative value to the energy for each active assignment. In contrast to [25], which uses a constant value, we utilize a variable value that models the confidence of occlusion at an image location. Thereby a strong decrease in depth at an image location $I_d(p) \gg I'_d(p)$ is an indicator of occlusion with a closer object. Using the weights $w_I(p) = I_d(p) - \min(I_d(p), I'_d(p))$ and $w_{I'}(p) = I'_d(p) - \min(I_d(p), I'_d(p))$ normalized over the full image to the range $[0, 1]$, we set

$$E^{occ}(x) = \sum_{a=(p,p') \in A} - \ (1 - \frac{w_I(p) + w_{I'}(p')}{2}) \ x_a \ . \tag{4}$$

In general, finding the global minimum of the energy function Eq. (1) is an NP-hard problem. However, the *Dual Decomposition Graph Matching* developed by Torresani et al. [25] finds a good approximative solution that is in practice often close to the global optimum, see Sec. 4.

3.2 Scene Flow

The result of the graph matching step is a set of sparse matches of descriptor segments. We use this result to get a dense 6D flow field by optimizing a discrete-domain energy. Extending the work from [15], we want the 6D scene flow $g : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from I to I' and g' from I' to I to minimize the energy

$$E(g, g') = \sum_{p \in \Omega \cup \Omega'} D(g_p^*) + \sum_{(p,q) \in N_4} V(g_p^*, g_q^*) + \sum_{p \in \Omega, p' \in \Omega'} C(g_p, g'_{p'}) \tag{5}$$

where $g_p^* \in \{g_p, g'_p\}$. The data term is from [15]:

$$\begin{aligned} D(g_p) = \sum_{H \in S_p} w(p, \pi(H))(\|\nabla I_c(\pi(H)) - \nabla I'_c(\pi'(g_p(H)))\|_2^2 \\ + \alpha \|g_p(H) - NN_{I'}(g_p(H))\|_2^2 \ ) \ , \end{aligned} \tag{6}$$

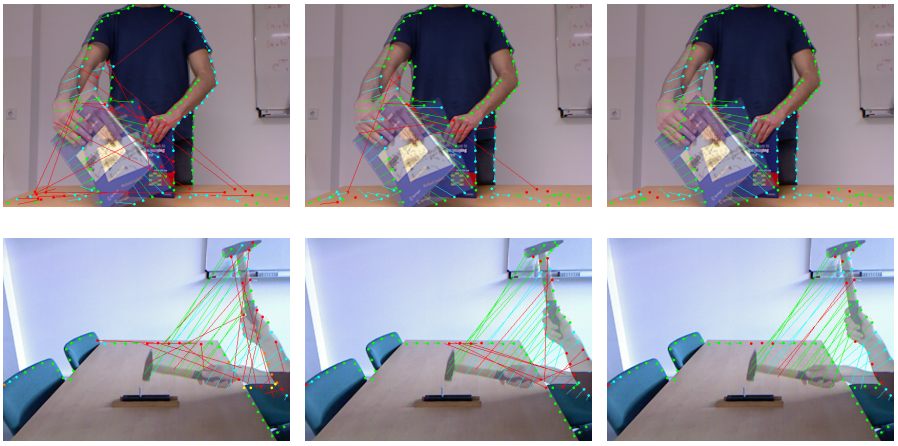
which measures RGB gradient constancy and geometric consistency of the scene flow g, for all 3D points H in a sphere S_p around the 3D back-projection of pixel p. Here ∇I_c is the image gradient, $\pi(H)$ is the projection of H onto image plane I, $NN_I(H)$ is the nearest neighbor of H in 3D back-projection of I, $\alpha > 0$ is a weighting constant and $w(p, p') = exp(-||I(p) - I(p')||_2/\gamma)$ is an adaptive support weighting [32]. The pairwise smoothness term

$$V(g_p, g_q) = \beta \ ||g_p(\bar{P}) - g_q(\bar{P})||_2^2 \tag{7}$$

is also similar to to [15]. Weighted with $\beta > 0$, it enforces smoothness by applying the motion of pixels p and q in the 4-connected neighborhood N_4 to the middle point $\bar{P} = \frac{1}{2}(P+Q)$ of their 3D back-projections P, Q. Additionally, we introduce the term $C(g_p, g'_{p'})$ which enforces consistency between forward and backward scene flow by penalizing the deviation from the "starting point"

$$C(g_p, g'_{p'}) = \begin{cases} ||p - \pi(g'_{p'}(g_p(P)))||_2 & \text{if } p' = \pi'(g_p(P)) \\ 0 & \text{otherwise} \end{cases} . \tag{8}$$

We minimize this energy in *three phases*. In the first phase, we obtain 6D rigid body motions for all sparse matches by mapping the corresponding 3D points and their surface normals into one-another such that the rotation is minimal [15]. Afterwards, pixels without an associated sparse match are assigned the 6D motion of their geodesically closest matched point with respect to the depth map. In the second phase, we use the PatchMatch variant of Hornacek et al. [15] to minimize the data term D only. The smoothness term V and consistency term C are in this phase only optimized implicitly by considering proposals from spatial neighbors and potential matches of the forward/backward scene flow, respectively. For the third phase, we cluster the 6D motions of the second phase that satisfy $C(g_p, g'_{p'}) < \tau$ with $\tau = 1$ into $K = 50$ clusters using the K-means clustering of the corresponding Rodriguez representations. The clustered



(a) Hungarian (b) Torresani at al. [25] (c) Ours

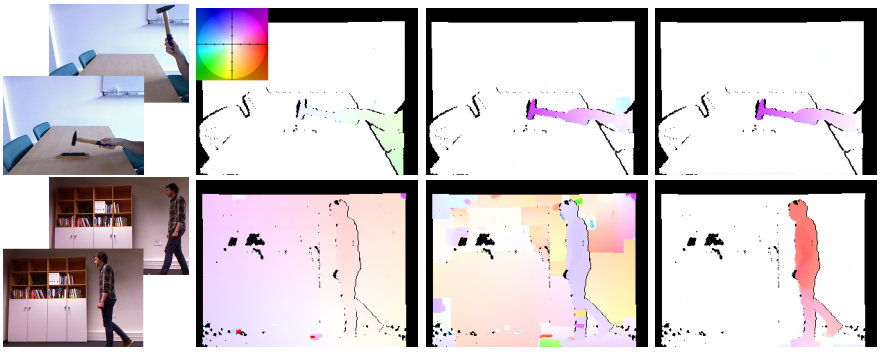
Fig. 2: **Visual comparison of graph matching results**. For illustration purpose both RGB images are super-imposed. Green means a correct match (or occlusion), blue is an almost correct match (definition in text) and red is a wrong match. Our result is clearly superior to the other techniques.

6D motions from this step serve as scene flow proposals for a global minimization of the energy in Eq. (5) via alpha expansion [5]. Here each expansion move runs QPBO [13], since the move-energy can be non-submodular. The alpha expansion is initialized with the motions from the second phase. Note, in contrast to [15] we optimize one global energy which includes consistency, smoothness, and appearance. This leads to considerably improved results, see Sect. 4.

4 Experiments

We recorded a dataset of seven RGB-D image pairs with the MS Kinect V1 camera.[1] Each image pair captures objects that undergo very large motion – see examples in Fig. 2. We present both quantitative results for graph matching, as well as qualitative evaluation of the final dense scene flow. In order to quantify the graph matching results, we use our algorithm to build description segments on image pairs. Then we generate ground truth matching by manually labeling each description segment in one image with its best corresponding description segment in the other image, or marking it as occlusion. Note that by construction, description segments centers in two images may not correspond exactly to the same physical 3D points. Therefore, we define *almost correct* matches to be those which are within the radius $r_{max} = 30$ of the correct description segments centers, see Fig. 1. Given ground truth matches, we can assign three class labels

[1] Available on our web page
http://cvlab-dresden.de/research/image-matching/graphflow/



(a) (b) (c) (d)

Fig. 3: **Comparison to SphereFlow [15]** for the sequence *Hammer* and *Walking*. Only the flow field of the left image is shown. (a) Original image pair. (b) Result of SphereFlow, where black pixels are unobserved depth values. For visualization, the magnitude of flow vectors is multiplied by 100. For these two sequences the result is an almost constant small motion everywhere. (c) Result of SphereFlow with graph matching points added to the pipeline of [15], before PatchMatch is applied. Only for the *Hammer* sequence gives better results, where the "pink-colored motion" on the hammer points towards the "up-right" direction (see flow color encoding). The *Walking* sequence is still degenerate (d) Our result. In both cases the motion estimation looks visually pleasing.

to each description segment matched by the graph matching algorithm: a) correct match; b) almost correct match and c) wrong match. Exemplary results are shown in Fig. 2. A full quantitative evaluation is given in Table 1. The percentages of correct matches are shown first, whereas the almost correct matches are denoted in brackets. We compare our graph matching with conventional nearest neighbor (SIFT) matching and Hungarian matching of the description segments that admits at most one-to-one matching [24]. Both methods do not exploit any graph structure and have a considerably lower performance. We also compare ours to the graph matching approach of Torresani et al. [25]. Both approaches use the same appearance term and optimization method, but different geometry and occlusion terms. In all but two cases our method outperforms the results of Torresani et al. [25] and a gain of up to 24.4% can be observed in the *Tea* scene. Finally, we evaluated the impact of our weighted occlusion term by fixing it to a constant value, denoted by "Const. Occ.". For nearly all scenes variable occlusion weights result in better performance.

As our final aim is the computation of scene flow, we are not only interested whether matches are correct or wrong, but also in the Euclidean distance between wrong matches and correct ones. These distances are shown in Table 2. Our approach has the smallest average error distance of all methods.

To better understand the optimization process, we finally analyze the energy of the dual decomposition framework with respect to the computed lower bounds

Sequence	SIFT	Hungarian [24]	Torresani [25]	Const. Occ.	Our
Board	63.3 (80.0)	70.0 (91.7)	70.0 (93.3)	**85.0 (95.0)**	**85.0 (95.0)**
Books	38.9 (64.3)	46.0 (71.4)	**59.5** (80.2)	55.6 **(83.3)**	**59.5 (83.3)**
Dinner	80.0 (89.5)	86.7 (93.3)	**88.6 (97.1)**	87.6 **(97.1)**	87.6 (96.2)
Hammer	34.6 (42.0)	51.9 (65.4)	59.3 (76.5)	65.4 (76.5)	**67.9 (81.5)**
Party	64.6 (86.5)	57.3 (81.3)	63.5 (88.5)	66.7 (90.6)	**67.7 (92.7)**
Tea	54.4 (90.0)	73.3 (85.6)	55.6 (88.9)	78.9 (91.1)	**80.0 (93.3)**
Walking	54.2 (72.9)	66.7 (81.3)	79.2 **(85.4)**	**83.3** (83.3)	**83.3** (83.3)

Table 1: **Quantitative evaluation of the graph matching results**. The percentages of correct matches and almost correct matches (in brackets). Our algorithm consistently outperforms a naive approach as well as a comparable approach [25] by up to 24.4% for the *Tea* sequence. Here "Const. Occ." means our full energy with a constant occlusion term. Best results are marked in bold.

Sequence	SIFT	Hungarian [24]	Torresani [25]	Const. Occ.	Our	Lower	Upper
Board	17.32	11.00	7.32	**3.78**	**3.78**	-2262.7	-2262.7
Books	47.90	36.75	15.03	7.50	**6.93**	-4170.6	-4103.8
Dinner	16.52	9.72	3.71	2.26	**2.24**	-3602.5	-3602.5
Hammer	56.77	33.62	29.75	23.11	**21.81**	-2101.4	-2097.8
Party	12.33	20.39	9.49	6.92	**6.41**	-3631.0	-3629.9
Tea	11.79	9.56	9.05	4.12	**3.69**	-3126.9	-3126.9
Walking	21.01	15.89	8.15	**7.66**	**7.66**	-1625.8	-1625.8

Table 2: **Quantitative evaluation of the graph matching results and associated optimization problem.** (Left) Average Euclidean distances of wrong matches as compared to ground truth. Overall, our matching results have equal or smaller error compared to all other methods. (Right) Lower and upper bound of the graph matching energy, where we reach in four out of seven cases global optimality.

for each scene, see Table 2 right. In four cases we reach global optimality while in the other three cases the lower bound is relatively tight. Given those results on our dataset, we conclude that our model seems to be a close approximation to an optimal energy formulation.

Dense 6D Scene Flow. We evaluate our method for obtaining dense scene flow, from the graph matches, on the same RGB-D scenes as they contain large displacements and untextured regions. Datasets with similar challenging data and ground truth scene flow are not available. Therefore, we restrict ourselves to qualitative evaluation. Our dataset is publicly available for future comparisons. We compare to SphereFlow [15], which is the state-of-the-art for large displacement scene flow estimation, as shown in their work. While for our scenes the result of SphereFlow is often decent, we observed that it sometimes returns a degenerate result where all motions are close to zero [2] – see examples in Fig. 3. The reason for such degenerate results is two-fold: First, SphereFlow [15] relies on SURF matches that are only available in textured areas. While our graph

[2] Adjusting the weighting parameters of [15] did not improve the results.

(a) time = 0 (b) time = 0.33 (c) time = 0.66 (d) time = 1

Fig. 4: **Visualizing our results as a 3D point cloud from a slightly different viewpoint**. (a-d) We warp the point clouds of the left and right images to generate intermediate images for different time points. Note that white pixels are due to missing depth measurements. We refer to our web page for a video.

matching provides good matches even in the absence of texture. Second, [15] does not optimize one energy but rather does a sequence of consistency checks before optimizing an energy with a smoothness term and a simplified data term. In contrast, we optimize an energy that includes a consistency term, which results in better and more stable solutions, see Fig. 3 (d). Additional results can be found on our web page.

To further evaluate the accuracy of our method, we interpolate between the image pairs using the estimated scene flow and render intermediate frames from a slightly different viewpoint, see Fig. 4. The resulting interpolated videos are realistic and show smooth transition.

5 Conclusion and Future Work

We propose to use graph matching of depth edges to estimate sparse, large displacement motions between two RGB-D images. Combining this with a new continuous-label energy for dense 6D scene flow, we are able to achieve state-of-the-art results. In a next step we will add additional fine-tuning, e.g. gradient descent, on top of the discrete optimization. In a broader context, this work may inspire new directions for optical flow estimation from RGB images, since depth edges are the locations in the image which are most challenging for correspondence search, and at the same time often the most important locations when creating new visual effects.

References

1. Alvarez, L., Deriche, R., Papadopoulo, T., Sánchez, J.: Symmetrical dense optical flow estimation with occlusions detection. In: Proc. ECCV. Springer (2002)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: PatchMatch: a randomized correspondence algorithm for structural image editing. TOG 28(3) (2009)
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proc. ECCV. Springer (2006)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. TPAMI 24(4) (2002)
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI 23(11) (2001)
6. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. TPAMI 33(3) (2011)
7. Canny, J.: A computational approach to edge detection. TPAMI 8(6) (1986)
8. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI 18(03) (2004)
9. Ferstl, D., Riegler, G., Ruether, M., Bischof, H.: CP–Census: A novel model for dense variational scene flow from RGB-D data. In: Proc. BMVC (2014)
10. Foggia, P., Percannella, G., Vento, M.: Graph matching and learning in pattern recognition in the last 10 years. IJPRAI 28(01) (2014)
11. Grzegorzek, M., Theobalt, C., Koch, R., Kolb, A.: Time-of-Flight and Depth Imaging. Sensors, Algorithms and Applications. Springer (2013)
12. Hadfield, S., Bowden, R.: Scene particles: Unregularized particle-based scene flow estimation. TPAMI 36(3) (2014)
13. Hammer, P., Hansen, P., Simeone, B.: Roof duality, complementation and persistency in quadratic 0–1 optimization. Mathematical Programming 28(2) (1984)
14. Herbst, E., Ren, X., Fox, D.: RGB-D flow: Dense 3-d motion estimation using color and depth. In: Proc. ICRA. pp. 2276–2282. IEEE (2013)
15. Hornácek, M., Fitzgibbon, A., Rother, C.: Sphereflow: 6 DoF scene flow from RGB-D pairs. In: Proc. CVPR. IEEE (2014)
16. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Proc. ICCV. IEEE (2007)
17. Jaimez, M., Souiai, M., Gonzalez-Jimenez, J., Cremers, D.: A primal-dual framework for real-time dense RGB-D scene flow. In: Proc. ICRA. IEEE (2015)
18. Leordeanu, M., Zanfir, A., Sminchisescu, C.: Locally affine sparse-to-dense matching for motion and occlusion estimation. In: Proc. ICCV. IEEE (2013)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2) (2004)
20. Quiroga, J., Devernay, F., Crowley, J.: Local/global scene flow estimation. In: Proc. ICIP. IEEE (2013)
21. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: Proc. ECCV. Springer (2010)
22. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. arXiv:1501.02565 (2015)
23. Sellent, A., Ruhl, K., Magnor, M.: A loop-consistency measure for dense correspondences in multi-view video. Image and Vision Computing 30(9), 641 – 654 (2012)

24. Smith, D.K.: Network flows: Theory, algorithms, and applications. Journal of the Operational Research Society 45(11) (1994)
25. Torresani, L., Kolmogorov, V., Rother, C.: A dual decomposition approach to feature correspondence. TPAMI 35(2) (2013)
26. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: Proc. ICCV. vol. 2, pp. 722–729. IEEE (1999)
27. Verhagen, B., Timofte, R., Van Gool, L.: Scale-invariant line descriptors for wide baseline matching. In: Proc. WACV. IEEE (2014)
28. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: Proc. ICCV. IEEE (2013)
29. Wang, Y., Zhang, J., Liu, Z., Wu, Q., Chou, P., Zhang, Z., Jia, Y.: Completed dense scene flow in RGB-D space. In: Proc. ACCV Workshops. pp. 191–205. Springer (2014)
30. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proc. ICCV. IEEE (2013)
31. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. TPAMI 34(9) (2012)
32. Yoon, K.J., Kweon, I.S.: Locally adaptive support-weight approach for visual correspondence search. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 924–931. IEEE (2005)
33. Zhang, L., Koch, R.: An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. Journal of Visual Communication and Image Representation 24(7) (2013)