Kinecting the dots: Particle Based Scene Flow From Depth Sensors

Simon Hadfield Richard Bowden
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, Surrey, UK, GU2 7XH
{s.hadfield,r.bowden}@surrey.ac.uk

Abstract

The motion field of a scene can be used for object segmentation and to provide features for classification tasks like action recognition. Scene flow is the full 3D motion field of the scene, and is more difficult to estimate than it's 2D counterpart, optical flow. Current approaches use a smoothness cost for regularisation, which tends to oversmooth at object boundaries. This paper presents a novel formulation for scene flow estimation, a collection of moving points in 3D space, modelled using a particle filter that supports multiple hypotheses and does not oversmooth the motion field. In addition, this paper is the first to address scene flow estimation, while making use of modern depth sensors and monocular appearance images, rather than traditional multi-viewpoint rigs. The algorithm is applied to an existing scene flow dataset, where it achieves comparable results to approaches utilising multiple views, while taking a fraction of the time.

1. Introduction

Scene flow is the 3 dimensional motion field of an observed scene, as opposed to optical flow which is the projection of this field onto the image plane. It is difficult to estimate scene flow, as observations on image planes are highly ambiguous. In this paper, inspiration is taken from particle filtering, and the problem is solved using a competing collection of scene point hypotheses, combined with a modern depth sensor.

Most current approaches to scene flow estimation assume a stereo, or multi-view camera system, in which the scene structure and motion is estimated simultaneously. To date there have been few attempts at solving scene flow estimation using direct depth sensors (such as time-of-flight cameras, or structured light). In this paper a Microsoft Kinect$^{\text{TM}}$system is used to provide extremely accurate scene structure, and three dimensional motion is estimated from only a single image sequence.

When estimating motion fields over a sequence, the previous estimate is often used to initialise the next frame. Thus a poorly estimated frame can have a negative impact on the rest of the sequence. The approach adopted in this paper, allows multiple hypotheses to be maintained, reducing this accumulation of errors. Additionally, oversmoothing of the structure and motion fields is avoided, which is the primary source of errors in contemporary scene flow algorithms.

1.1. Related Work

The most common approach to estimating scene flow is to perform an optimisation on a global energy function, including brightness constancy matching, and some regularisation. Some authors add additional elements to the energy function such as estimating the camera extrinsics [19] or additional constraints on stereo matching [22]. This optimisation approach is generally slow, although Rabe *et al*. achieved real time performance by using a gpu implementation [14]. The algorithm presented in this paper is demonstrated to operate much faster than previous, non-parallelised, implementations.

The regularisation required to constrain these systems, causes oversmoothing of discontinuities (such as object boundaries) in both the structure and motion estimate. It is possible to reduce this effect, by segmenting the input images and applying smoothness constraints only within segments [10], however, such segments suffer projective distortions when being compared in a multi-camera setup, and so the brightness constancy matching is less accurate.

Basha *et al*. showed [1] that estimation could be improved, by formulating the problem as a point cloud in 3D space, rather than the commonly used 2D parameterisations, where smoothness constraints are less applicable. This representation allowed the system to be easily applied to any number of cameras, in any setup.

A number of authors [12, 8, 4] make use of 3D formulations, based on meshes rather than point clouds. However this approach limits the possible motions and structure, making it appropriate for some applications, but less generally applicable than using point clouds.

Most approaches estimate dense motion fields, however Devernay *et al*. performed a sparse scene flow estimation [6]. The motion estimates were obtained from tracking surfels, originally proposed by Carceroni and Kutulakos [3]. This leads to a tradeoff between precision and coverage in the estimation.

The majority of previous work operates in either a stereo or multi-view setup. Some authors [9, 21] attempt to use state of the art depth reconstruction algorithms, to provide or initialise the structure underlying their motion field. However, no work has previously been done making use of direct depth sensors, or estimating scene flow from monocular appearance sequences, which is the focus of this work.

Spies *et al*. [18] incorporated a depth sensor into scene flow estimation, by extending the constraints from the appearance domain. This allowed very sparse motion estimation to be performed, which was then regularised to fill unestimated regions. Lukins and Fisher [11] then investigated the use of various colour spaces combined with depth streams, and Schuchert *et al*. [16] investigated performance in the presence of illumination changes.

The work in this paper is inspired, in part, by that of Davison *et al*. where Simultaneous Location and Mapping (SLAM) was performed with monocular sequences, by spreading depth hypotheses along viewing rays and evaluating them in subsequent frames [5]. This was used to estimate structure sparsely, rather than the dense motion field, and multiple hypotheses were only maintained briefly. Conversely, the algorithm presented here estimates the motion field using a cloud of velocity hypotheses at each position in the scene.

The work of Vedula *et al*. [20] also bears some similarity to the techniques proposed in this paper. Voxel colorisation was used to examine a coarse version of all possibilities in the scene space, and find consistent regions based on background subtraction. Ruttle *et al*. [15] later expanded this work, exploring the use of additional heuristics.

1.2. Paper Structure

In the rest of the paper, section 2 discusses general multi-sensor scene flow estimation from a probabilistic viewpoint, and relates it to the use of modern depth sensors. Section 3 describes the Scene Particle approach to performing this estimation. The coverage of the estimated motion field is discussed in section 3.2, and a variant of the algorithm is presented which ensures dense estimation. In section 4.1 qualitative evaluation of the system is performed on a sequence recorded with a Kinect$^{\mathrm{TM}}$. Quantitative results are given in section 4.2, where the algorithm is applied to an existing scene flow dataset, and finally future directions are discussed, along with the conclusions of the paper in section 5.

2. Scene Probability Space

When estimating a dense motion field, the intention is to find the best 3D velocity vector $\mathbf{v}$, for each structural position $\mathbf{r}$. The scene probability space defines, for every combination of structure point and motion vector, the probability of existing in the scene. Given a set of observations $\mathbf{i}$, it is possible to represent the posterior probability $\mathbf{p}(\mathbf{r},\mathbf{v}|\mathbf{i})$ in terms of the likelihood and the prior probability distributions.

$$\mathbf{p}(\mathbf{r},\mathbf{v}|\mathbf{i}) \propto \mathbf{p}(\mathbf{i}|\mathbf{r},\mathbf{v})\mathbf{p}(\mathbf{r},\mathbf{v}) \tag{1}$$

The prior probability $\mathbf{p}(\mathbf{r},\mathbf{v})$ can be obtained from the posterior at the previous frame, in combination with a motion model. The likelihood $\mathbf{p}(\mathbf{i}|\mathbf{r},\mathbf{v})$ is formulated using 2 separate terms, as the observations $\mathbf{i}$ include information from both appearance and depth sensors.

The first likelihood term $\mathbf{g}(\mathbf{i}|\mathbf{r},\mathbf{v})$ is formulated using the brightness constancy assumption, found in most optical flow and scene flow approaches. This assumption states that the intensity of a world point is identical when viewed from any angle. With a set of M cameras in any multi-view setup, the input observation $\mathbf{i}$ is the set images $I_{1..M}$. If the cameras have projection matrices $\mathbf{\Pi_{1..M}}$, any true world point $\mathbf{r}$ in the scene will satisfy the condition:

$$\sum_{m=0}^{M}\sum_{q=0}^{M} |I_q\left(\mathbf{\Pi_q r}\right) - I_m\left(\mathbf{\Pi_m r}\right)| = 0 \tag{2}$$

If each camera produces an image sequence of T frames $I_{1..M}^{1..T}$, then a scene point $\mathbf{r}$ with 3D velocity $\mathbf{v}$ can be related between cameras and frames by the condition:

$$\sum_{m=0}^{M}\sum_{q=0}^{M} |I_q^t\left(\mathbf{\Pi_q r}\right) - I_m^{t-1}\left(\mathbf{\Pi_m}\left(\mathbf{r}-\mathbf{v}\right)\right)| = 0 \tag{3}$$

These conditions generalise to any number of cameras, in any setup. In contrast, the image plane and disparity based formalisation often used, generally requires rectified images and a parallel camera setup.

The Kinect$^{\mathrm{TM}}$system contains one depth camera (with projection matrix $\mathbf{\Pi_d}$, producing image sequence $I_d^{1...T}$) and one appearance camera (with projection matrix $\mathbf{\Pi_a}$, producing image sequence $I_a^{1...T}$). Positions in the scene ($\mathbf{r}$) are obtained by backprojecting values from the depth images, using the reverse projection matrix $\mathbf{\Pi_d}^{\mathbf{-1}}$. The divergence from equation 3 at each point in the scene probability space is defined as the cost $\mathbf{c}(\mathbf{i}|\mathbf{r},\mathbf{v})$, with higher costs indicating reduced intensity matching.

$$\mathbf{c}(\mathbf{i}|\mathbf{r},\mathbf{v}) = \| \mathbf{I_a^t}\left(\mathbf{\Pi_a \Pi_d^{-1} I_d^{t\text{-}1}}\right) - \mathbf{I_a^{t\text{-}1}}\left(\mathbf{\Pi_a}\left(\mathbf{\Pi_d^{-1} I_d^{t\text{-}1}} - \mathbf{v}\right)\right) \| \tag{4}$$

Finally the first likelihood term $\mathbf{g}(\mathbf{i}|\mathbf{r},\mathbf{v})$ is determined from $\mathbf{c}(\mathbf{i}|\mathbf{r},\mathbf{v})$ using equation 5 with $\epsilon = 0.001$, which is a smooth approximation of L_1 (see [2]).

$$\mathbf{g}(\mathbf{i}|\mathbf{r},\mathbf{v}) = \frac{\mathbf{1}}{\sqrt{\mathbf{c}(\mathbf{i}|\mathbf{r},\mathbf{v})^{\mathbf{2}} + \epsilon^{\mathbf{2}}}} \tag{5}$$

The second element $\mathbf{d}(\mathbf{i}|\mathbf{r},\mathbf{v})$ of the likelihood, is called the structure conformancy term, and incorporates information from the depth sensor at the previous frame. Each position $\mathbf{r}$ in the distribution is flowed backwards by it's motion estimate $\mathbf{v}$, and the distance is calculated to the closest point $\mathbf{r^{t-1}}$ from the previous scene structure.

$$\mathbf{d}(\mathbf{i}|\mathbf{r},\mathbf{v}) = \mathbf{min_m}((\mathbf{r^t_n} - \mathbf{v^t_n}) - \mathbf{r^{t-1}_m}). \tag{6}$$

To more heavily favour close conformance, the exponential decay of $\mathbf{d}(\mathbf{i}|\mathbf{r},\mathbf{v})$ is used, to produce the likelihood as in equation 7.

$$\mathbf{p}(\mathbf{i}|\mathbf{r},\mathbf{v}) = \mathbf{g}(\mathbf{i}|\mathbf{r},\mathbf{v})\mathbf{e}^{-(\mathbf{d}(\mathbf{i}|\mathbf{r},\mathbf{v})))} \tag{7}$$

3. Scene Particle Algorithm

The maintenance of these high dimensional, continuous probability distributions is obviously intractable. Instead the Scene Particle algorithm represents the posterior distribution $\mathbf{p}(\mathbf{r},\mathbf{v}|\mathbf{i})$ as a population of N weighted particles. Each particle p_n has an associated weight w_n, and is represented by a 6D vector consisting of the 3D position in the world $\mathbf{r}$, and a 3D motion vector $\mathbf{v}$, and is referred to here as a Scene Particle. Many Scene Particles may have the same spatial position $\mathbf{r}$, while maintaining separate motion hypotheses at that scene position.

$$p_n = (\mathbf{r},\mathbf{v}) \in \mathbb{R}^6 \tag{8}$$

A resampling stage is performed after each new observation. Each particle spawns a number of copies, which are diffused by Gaussian noise. For a population of N particles, the number of children spawned by particle p_n is $w_n \times N$. Thus, areas of the scene space with high probability will contain more particles, while areas of low probability will become sparse. The drawback of this resampling is that only a portion of the positions obtained from the depth sensor will have Scene Particles estimating their motion. In section 3.2 a variation of the algorithm ensuring fully dense flow estimation is discussed.

It is important to note that in most particle based systems, each particle provides a complete solution to the task. However, in the Scene Particle algorithm, a particle provides only a single element of the motion field. To obtain an estimated flow field from the Scene Particle population, the weighted average of all Scene Particles at each position $\mathbf{r}$, is calculated.

3.1. Iterative Estimation

As with several other scene flow approaches [9, 13], the Scene Particle algorithm employs a coarse to fine strategy, to reduce the effects of local maxima in the probability distribution. The input images from each view are converted to a scale pyramid, each level is applied in turn to the Scene Particles, as a new observation. The number of Scene Particles remains constant at all times, consequently smaller images have more hypotheses per camera ray. Thus, each level refines the previous estimate to finer spatial scale, with fewer hypotheses per ray.

In addition to this coarse to fine approach, a second series of iterations is performed at each scale. The resampled particles are diffused in a gaussian manner, and their weights are updated. After each of these inner iterations, the standard deviation of the diffusion Gaussian is halved. This allows particles to begin with more exploratory behaviour, and to then converge over time.

Within each iteration, Scene Particles are processed independently. As such, the algorithm is eminently suitable for a parallelised implementation (such as on a GPU), but such an implementation was not produced for the purposes of this paper.

3.2. Scene Coverage

Particle filtering systems have an unfortunate tendency to converge to the highest peak over extended periods (see [17]). In many applications this is acceptable, as the global maximum is desired. In the Scene Particle algorithm however, the required output comprises the set of all local maxima. The loss of Scene Particles representing these local maxima, increases the sparsity of the estimated motion field. Modifying the Scene Particle weight update equation, as in 9, counters this effect, as it reduces the contribution of information from previous iterations, based on the frame separation. This allows greater scene coverage to be maintained over prolonged sequences.

$$w^t_n = \frac{\mathbf{p}(\mathbf{i}|\mathbf{p_n}) + \mathbf{w^{t-1}_n}}{\sum\limits_{j=0}^{N}\left(\mathbf{p}(\mathbf{i}|\mathbf{p_j}) + \mathbf{w^{t-1}_j}\right)} \tag{9}$$

For most applications, this results in a semi-dense estimate, which is sufficient. For other situations, an additional modification termed Ray Resampling, is proposed. In the Ray Resampling scheme every Scene Particle is grouped to the closest ray from the camera (the pixel it projects to in the appearance image). Resampling of the Scene Particle population is then performed separately for the sub-population grouped onto each ray. Essentially this approach equates to using a separate particle filter for each ray/pixel, and allowing the motion models to move particles between filters during frame transitions. This ensures that every structure point from the depth sensor has it's motion estimated by the same number of Scene Particles.

4. Results

4.1. Kinect Sequence

Initial qualitative results were obtained on recordings from a Microsoft Kinect ^{TM}, accessed using OpenNI's software. Figure 1 shows the output of the sensors, and the resultant estimation. The motion field has been downsampled for ease of viewing. Original images were at 640x480 resolution, whereas the displayed motion field contains 128x96 flows. Because of this downsampling, the algorithm is able to operate without paralellisation in under 10 seconds on a standard desktop machine.

The depth input, figure 1.A, is the Z distance at each point from the depth sensor, reprojected to the appearance camera. Lighter regions are further from the sensor, while black regions could not be measured. Some unmeasured regions are due to areas of the appearance image which are occluded in the depth camera. The remaining unmeasured regions are due to reflective surfaces interfering with the depth sensor.

The motion field in figure 1.C, shows the velocity estimated at each structure point, as a flow line, starting at the cyan vertex and moving to the white vertex. The background in the scene is stationary, and as such has little motion estimated in any dimension. Plausible motion estimates can be seen on the leg and arm, with the fastest moving areas at the end of the leg and the foot.

Due to the single viewpoint used, it is possible to see the "shadow" projected by the foreground subject on the rest of the scene. Areas of background which were occluded in the previous frame, such as those along the bottom of the leg projection shadow, produce incorrect flow estimates. This is due to the structure conformancy cost (equation 6) favouring flows from neighbouring structure which was visible in the previous frame.

Examining Figure 1.C, it is obvious that the standard oversmoothing artifacts from regularized optimisation approaches are not present. Near occlusion boundaries, Scene Particles initial position attach to one object or the other, rather than averaging between them. Additionally Scene Particle velocities on each side of boundaries are able to maintain completely different directions. This is due to Scene Particles being examined in isolation, without requiring local consistency.

The scene flow estimation results, for the full 280 frame sequence, can be viewed at www.computing.surrey.ac.uk/personal/pg/S.Hadfield/sceneparticle

4.2. Middlebury Comparison

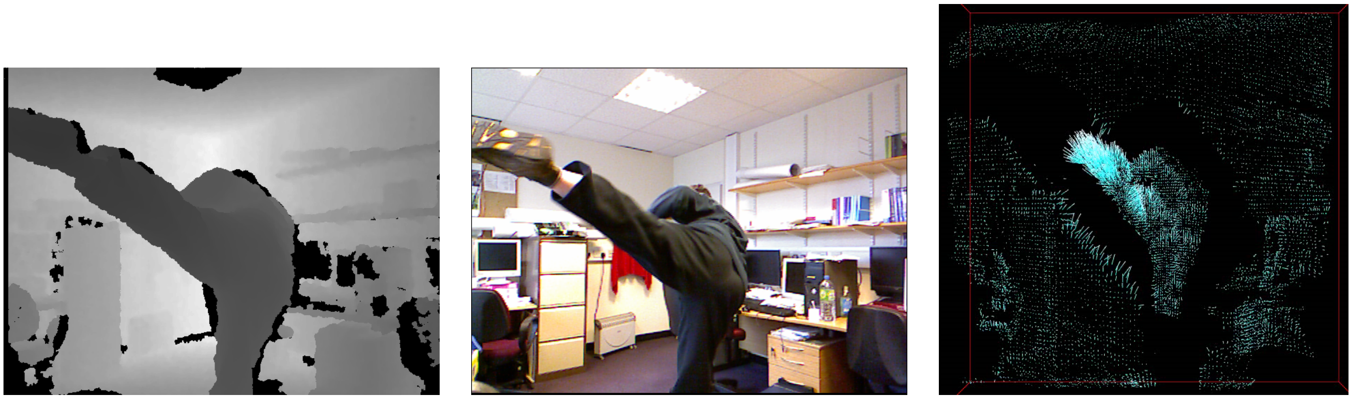
In order to perform a quantitative evaluation of the algorithm, an existing scene flow dataset was used. In [1], 8-camera stereo datasets from Middlebury are used to simulate scene flow. The 8 images are rectified, the cameras are parallel and equally spaced along the X axis. Using the images from camera 2 as the first frame, and camera 6 as the second frame, is equivalent to a stationary camera in a scene where every object moves along the x axis, with the same velocity. This allows ground truth motion at every point in a real, cluttered scene, facilitating comparisons with existing techniques.

Scene Particles explore motion within certain bounds (discussed further in section 4.3). This velocity range can be set to the maximum observable velocity of the cameras (based on the field of view and framerate), or can make use of application specific knowledge, when available. In the Middlebury datasets, the velocity at every position is identical (equal to the inverse of the camera velocity). However this knowledge is not exploited for these experiments. Velocities are explored between +/- half the maximum velocity in each scene. This range is explored in all three dimensions, despite no motion being present in Y and Z.

As in [1], parameters are estimated for the camera setup. The disparity images are used to simulate the output of the depth sensor (although they are far more coarsely quantised than the Kinect). Estimated scene flow is then projected back onto the image plane for comparison with the ground truth (which is provided in terms of pixels per frame). The motion field estimate at each pixel is taken as the weighted average of all Scene Particles projecting to that pixel. As an error measurement, the Normalised Root Mean Square (NRMS) Error is measured, as described in [1]. This quantifies the accuracy of estimated motion magnitude, scaled by the ground truth, to be comparable between datasets. In addition the Average Angular Error (AAE) is examined, to determine the accuracy of motion direction. Finally the listed coverage relates to the percentage of the input structure, represented by the motion field.

Table 1 shows the results, compared to the approaches of several other authors. Unlike the Scene Particle algorithm, these alternative approaches estimate scene structure, as well as scene flow, but require multiple viewpoints. Experiments were also performed, applying optical flow to the appearance images, and then using depth data to infer the 3D flows. Similar to the Scene Particle algorithm this uses a single viewpoint, and only estimates motion for a given structure. The Gunnar Farneback [7] optical flow algorithm was used, as implemented in OpenCV.

The Scene Particle algorithm consistently estimates motion magnitude, more accurately than previous approaches, despite making use of fewer views. This proves true for velocities in the image plane, and perpendicular to it. The



(A) (B) (C)

Figure 1. Qualitative images from the Kinect sequence. (A) The output of the depth sensor. (B) The output of the appearance sensor. (C) The motion vector field estimated, vectors start at the cyan end and move to the white end. Length of the vector indicates the magnitude.

Algorithm	Dataset	Views Used	Optical Flow NRMSE	Stereo Flow NRMSE	AAE (deg)	Coverage
Scene Particle	Cones	1	0.10	**0.00**	5.10	49%
Scene Particle + RR	Cones	1	**0.09**	**0.00**	5.02	100%
Op. Flow + Depth	Cones	1	0.22	0.38	4.63	88%
[1]	Cones	2	3.07	0.03	0.39	100%
[1]	Cones	4	1.32	0.01	**0.12**	100%
[9]	Cones	2	5.79	8.24	0.69	100%
Scene Particle	Teddy	1	**0.10**	**0.00**	5.10	50%
Scene Particle + RR	Teddy	1	0.11	**0.00**	5.04	100%
Op. Flow + Depth	Teddy	1	0.31	0.29	12.33	68%
[1]	Teddy	2	2.85	0.07	1.01	100%
[1]	Teddy	4	2.53	0.02	**0.22**	100%
[9]	Teddy	2	6.21	11.58	0.51	100%
Scene Particle	Venus	1	**0.08**	**0.00**	5.50	51%
Scene Particle + RR	Venus	1	0.09	**0.00**	5.44	100%
Op. Flow + Depth	Venus	1	0.38	0.23	12.21	98%
[1]	Venus	2	1.98	**0.00**	1.58	100%
[1]	Venus	4	1.55	**0.00**	1.09	100%
[9]	Venus	2	3.70	3.05	**0.98**	100%

Table 1. Results of Scene Particle motion estimation, using monocular appearance + depth. Compared with [1] and [9] using multiple views to estimate structure and motion. Also compared to optical flow estimation, incorporated with depth data. (RR is ray-resampled).

approach also outperforms the dedicated optical flow algorithm, showing that the incorporation of depth information at an earlier stage, allows more accurate flow estimates, even when reprojected to lie on the image plane.

Directional estimation accuracy is slightly lower than existing techniques. This is because, with a single view point, small perturbations of the motion vector often have no effect on the pixel projection of the particle. However, with more viewpoints, smaller deviations can be expected to affect the projection in at least one image, making it possible to distinguish between more finely separated motion hypotheses. The optical flow and depth approach also suffers due to this.

Performance of the standard algorithm is similar to that of the Ray Resampling variant. It might be expected that the Scene Particles in the standard algorithm would converge on areas of the scene with low ambiguity, and thus estimate accuracy would be higher, with reduced scene coverage. However this does not appear to be the case, with Ray Resampling actually producing more accurate directional estimates on all 3 datasets (albeit with slightly worse magnitude estimates).

The Scene Particle algorithm processed the middlebury dataset on a single core desktop machine in under 10 minutes, as opposed to 5 hours for [9] (run time was not reported in [1]). The optical flow and depth based approach requires 6 seconds.

4.3. Precision vs Exploration

Scene Particles exist within a certain volume of the velocity space. The smaller this search space is, the fewer Scene Particles are needed to explore it at a given spatial position. This means if application specific knowledge is

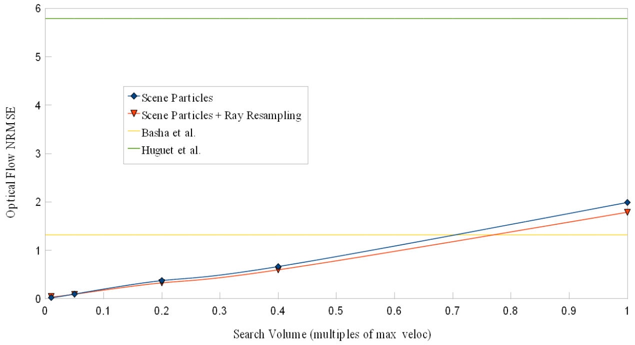


Figure 2. Velocity search volume (as a fraction of maximum velocity), against optical flow error, on the Cones dataset. [1] and [9] are benchmark lines for comparison.

available to reduce the search space, application speed can increase, at no cost to performance. Conversely by using such knowledge, while keeping the number of Scene Particles (and hence the run time) constant, improved accuracy can be obtained, as shown in figure 2.

With the number of particles used in the experiments, the Scene Particle algorithm outperforms existing methods, when exploring velocities up to 75% of the maximum visible velocity. Scene Particles moving at 75% maximum velocity would cover $\frac{3}{4}$ of the scene in a single frame, and so are unlikely to be visible in the following frame. Exploring velocities larger than this range would require additional particles and hence more computation, to maintain top performance (although the approach is still an order of magnitude faster). For a given search volume, the Ray Resampling approach slightly outperforms the standard algorithm, when using the same number of Scene Particles.

5. Conclusions

A novel Scene Particle approach to 3D motion estimation was proposed, and demonstrated to provide comparable performance to the current current state-of-the-art, at a fraction of the computational cost. The algorithm is also capable of operating on single viewpoint sequences unlike traditional approaches, making new applications viable. Furthermore, it is one of the few scene flow estimation system capable of making use of modern depth sensor technology such as the Kinect $^{\text{TM}}$, rather than relying on stereo matching algorithms.

Future work is planned to develop the Scene Particle algorithm for application in multi-view scenarios, including classic multi-view appearance datasets, but also multi-view depth and appearance data. It is expected that this will allow the estimation of the motion field direction to reach the levels of existing techniques making use of several views, while further increasing the accuracy of the magnitude estimate. Also under consideration is a parallelised implementation, further improving runtime, with a view to real time operation.

Acknowledgements

This work is supported by the European Community's Seventh Framework Programme (FP7/2007-2013) grant agreement no 231135 (Dicta-Sign).

References

[1] T. Basha, Y. Moses, and N. Kiryati. Multiview scene flow estimation: A view centered variational approach. In *CVPR*, 2010. 1, 4, 5, 6

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 3

[3] R. L. Carceroni and K. N. Kutulakos. Multi-view scene capture by surfel sampling: from video streams to non-rigid 3d motion, shape and reflectance. In *ICCV*, 2001. 2

[4] J. Courchay, J. Pons, P. Monasse, and R. Keriven. Dense and accurate spatio-temporal multi-view stereovision. In *ACCV*, 2009. 1

[5] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *PAMI*, 2007. 2

[6] F. Devernay, D. Mateus, and M. Guilbert. Multi-camera scene flow by tracking 3-d points and surfels. In *CVPR*, 2006. 2

[7] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *ICCV*, 2001. 4

[8] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. In *CVPR*, 2008. 1

[9] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 2, 3, 5, 6

[10] R. Li and S. Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. *CVIU*, 2008. 1

[11] T. Lukins and R. Fisher. Colour constrained 4d flow. In *BMVC*, 2005. 2

[12] J. Neumann and Y. Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *IJCV*, 2002. 1

[13] J. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV*, 2007. 3

[14] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *ECCV*, 2010. 1

[15] J. Ruttle, M. Manzke, and R. Dahyot. Estimating 3d scene flow from multiple 2d optical flows. In *IMVIP*, 2009. 2

[16] T. Schuchert, T. Aach, and H. Scharr. Range flow in varying illumination: Algorithms and comparisons. *PAMI*, 2009. 2

[17] H. Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequence*. PhD thesis, Stockholm Royal Institute of Technology, 2001. 3

[18] H. Spies, B. Jahne, and J. Barron. Range flow estimation. *CVIU*, 2002. 2

[19] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *ECCV*, 2010. 1

[20] S. Vedula, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, 2005. 2

[21] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense sceen flow from sparse of dense stereo data. In *ECCV*, 2008. 2

[22] Y. Zhang and C. Kambhamettu. On 3-d scene flow and structure recovery from multiview image sequences. *Trans. SMC*, 2003. 1