Scene Particles: Unregularized Particle Based Scene Flow Estimation

Simon Hadfield, *Member, IEEE,* Richard Bowden, *Senior Member, IEEE*

Abstract—In this paper, an algorithm is presented for estimating scene flow, which is a richer, 3D analogue of Optical Flow. The approach operates orders of magnitude faster than alternative techniques, and is well suited to further performance gains through parallelized implementation. The algorithm employs multiple hypothesis to deal with motion ambiguities, rather than the traditional smoothness constraints, removing oversmoothing errors and providing significant performance improvements on benchmark data, over the previous state of the art.

The approach is flexible, and capable of operating with any combination of appearance and/or depth sensors, in any setup, simultaneously estimating the structure and motion if necessary. Additionally, the algorithm propagates information over time to resolve ambiguities, rather than performing an isolated estimation at each frame, as in contemporary approaches.

Approaches to smoothing the motion field without sacrificing the benefits of multiple hypotheses are explored, and a probabilistic approach to Occlusion estimation is demonstrated, leading to 10% and 15% improved performance respectively.

Finally, a data driven tracking approach is described, and used to estimate the 3D trajectories of hands during sign language, without the need to model complex appearance variations at each viewpoint.

Index Terms—Scene Flow, Scene Particles, Motion Estimation, 3D, 3D Motion, Particle, Particle Filter, Optical Flow, Hand Tracking, Sign Language, Tracking, Occlusion Estimation, Probabilistic Occlusion, Occlusion, Bilateral Filter, 3D Tracking, Motion Segmentation

1 Introduction

Optical flow is commonly used in many applications, and defines the dense motion field of points within an image. Scene flow extends this, incorporating the 3D structure of a scene, as well as it's three dimensional motion field (which can be projected onto the image plane to obtain the optical flow) as shown in figure 1. Estimating Scene flow is a challenging task, due to the ambiguity inherent in the observations. In this paper, an approach to scene flow estimation is proposed, which solves these ambiguities by propagating multiple hypotheses through time, and allowing future observations to resolve them.

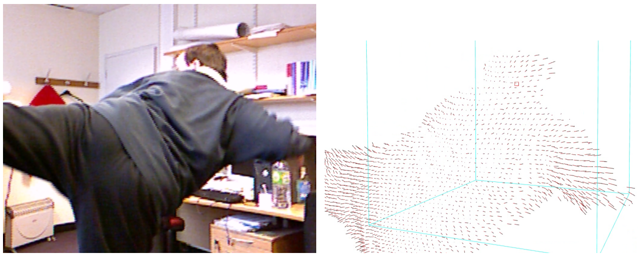


Fig. 1: An estimated 3D motion field, from a sequence of a person performing a kicking action. Motion vectors move from the red to purple vertices.

Estimating the motion field and structure provides a high level understanding of the scene, and can be valuable for a number of tasks such as segmentation [1], tracking [2], gesture recognition [3] and robot navigation [4]. The earliest approaches to estimating 3D scene motion was in the field of structure from motion. Such techniques tended to focus on monocular systems, and relied on matching sparse features between frames [5], [6], [7]. However, such approaches were often limited to rigidly-deforming scenes. Vedula *et al.* introduced scene flow estimation [8], [9], [10], using multiple viewpoints, to allow dense motion estimation in a freely deformable scene. More recently researchers have begun applying scene flow estimation techniques to depth sensors, known as "Range Flow" [11], [12], [13], [14]. The techniques proposed in this paper are applicable to both multi-view scene flow estimation, and combined appearance & depth sensor estimation.

Most current approaches to scene flow estimation are based on the optimization of an energy function, generally an extension of the optical flow constraint equation, which favors intensity matches between viewpoints and frames [15], [16]. However, the aperture problem, well documented in optical flow estimation research, also occurs when estimating scene flow. The most common approach to solving this ambiguity is to introduce another constraint, as a regularization term in the energy function, favoring smoothness of the motion field [17], [18], [19]. However, this regularization produces over-smoothing artifacts at

- *S. Hadfield and R. Bowden are with the Centre for Vision Speech and Signal Processing, at the University of Surrey, Guildford, England, GU2 7XH.*
E-mail: {S.Hadfield, R.Bowden}@surrey.ac.uk

discontinuities.

In [20] a preliminary version of the Scene Particles algorithm was presented, operating on RGB-D image sensors (specifically Microsoft Kinect $^{\text{TM}}$). In this paper the approach is extended to a general framework, capable of using any combination of appearance and depth sensors in any setup. Additionally, methods for incorporating probabilistic occlusion awareness, and artifactless motion field smoothing are discussed. Finally we provide a more comprehensive analysis of the algorithms properties, including robustness to noise, propagation of information and sparsity.

The paper is organized as follows. Initially an overview of current scene flow estimation techniques is provided in section 2. Next, a probabilistic formalization of scene flow estimation with any collection of input sensors is presented in section 3. This is then related to the discrete, particle based, approach, during section 4. Sparsity issues are discussed in section 4.2, and approaches to occlusion awareness and motion smoothing are presented in sections 4.3 and 4.4 respectively. Section 6.1 compares the accuracy of the algorithm to other contemporary approaches, in both multimodal and appearance only settings. The performance on longer sequences and the benefits of information propagation are discussed in section 6.2, and the various smoothing schemes are explored quantitatively in section 6.3. The performance of the algorithm in relation to sampling sparsity and sensor noise is explored in sections 6.4 and 6.5 respectively. Section 7 presents an example application, using the Scene Particle algorithm to calculate hand trajectories in 3D during sign language. Finally conclusions are drawn in section 8.

2 Related Work

As mentioned, many previous techniques for estimating scene flow have focused on optimization approaches, incorporating smoothness constraints on both the structure and motion. However, some authors [21], [22], [23] avoid this, by tracking only a sparse number of feature points, relating to the vertices of a known mesh. The dense structure and motion of the scene can then be interpolated along the mesh. In these approaches there is no smoothing over object boundaries, as discontinuities are inherently modeled. However, an initialization is required to compute the mesh topology, and motions leading to topology changes can generally not be handled. Devernay *et al.* take this a step further [24] and estimate a sparse scene flow only at the tracked surfels (originally proposed by Carceroni and Kutulakos [25]) without fitting a mesh. This removes the need for initialization but provides a very sparse estimate.

Alternatively, a number of techniques have been proposed for employing standard, optimization based, estimation, while mitigating the oversmoothing effects. One of the simplest [15], [16], relies on reducing the weighting of the regularization term, based on local image gradients. This is intended to reduce smoothing at object boundaries, but can lead to poor performance in highly textured regions.

Another approach, employed by Zhang *et al.* [26] (among others [27], [28]), is to remove the smoothness term from the optimization, and instead to define the data matching term to incorporate a neighborhood around each pixel. Such approaches are useful, as the region of oversmoothing at discontinuities is limited by the neighborhood size. A related approach is presented in [29] employing image segmentation, and estimating consistent motions within each segment, while not enforcing smoothness between segments. Matching segments in this manner assumes that the surface normal is roughly constant between views and between frames, implying the cameras are in a near parallel setup, and the scene exhibits little rotational motion. These assumptions about the camera setup are later loosened in [30], [31] by fitting the parameters of a motion model to each segment. Assuming the segmentation correctly finds discontinuities, this approach has the potential to eliminate over-smoothing artifacts altogether, however if segments become too small (as in highly textured regions), they may not contain enough information to be unambiguous.

The techniques most closely related to the Scene Particle algorithm, are those based on voxel colorization. This was initially developed by Vedula *et al.*[9], [10], and focuses on the brute force analysis of possible motions. Due to the inherent complexity, such approaches often explore a coarsely quantized space in order to remain tractable, resulting in a loss of fidelity. Ruttle *et al.*[32] reduced the complexity of the approach by introducing a number of additional heuristic constraints. In contrast, the Scene Particle algorithm operates in the original continuous space, using a collection of discrete samples.

More recently Basha *et al.*[33] developed a highly scalable, voxel based approach, where structure selection is deferred, until motion has been estimated. This bears some similarities to the ability of the Scene Particle algorithms to avoid making a hard decision as to the single "best" motion estimate (although no method is provided for utilizing this information during future estimates). The Scene Particle algorithm achieves this by maintaining all hypothesis, with associated probabilities, between frames. Ambiguities can then be resolved via the accumulation of observations, rather than by relying on smoothness assumptions to select the single smoothest hypothesis at each frame.

3 Probabilistic Scene Flow

For a given scene, there is a 3 dimensional set of possible structure points. A particular location within this continuous space is referred to with the vector $\mathbf{r} = \{x, y, z\}$. In addition, at every possible $\mathbf{r}$, there is also a continuous 3 dimensional space of possible motions. The vector $\mathbf{v}$ refers to one possible velocity from this space ($v = \{v_x, v_y, v_z\}$). The goal of scene flow estimation, is to extract the most probable set of structure points $\mathbf{r}$ and corresponding motions $\mathbf{v}$, given a set of input observations $\mathbf{i}$. To this end, the 6 dimensional Scene Probability distribution $\mathbf{p}(\mathbf{r}, \mathbf{v}|\mathbf{i})$ is estimated, the peaks of which provide an estimate of the structure and motion field. The resulting estimate is not restricted to a single peak per optical ray, instead providing a multi-hypothesis estimate for every pixel. Note that "observations", in this paper, refer to the set of images obtained from a number of synchronized appearance and/or depth sensors, in a known but unrestricted configuration. In this work, we do not consider the case where the configuration of the sensors changes over time, however the technique directly extends to this situation as long as calibration information is available.

The Scene Probability distribution $\mathbf{p}(\mathbf{r}, \mathbf{v}|\mathbf{i})$ at any frame (i.e. for a particular set of observations), is estimated as follows.

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

The prior probability $\mathbf{p}(\mathbf{r}, \mathbf{v})$ is obtained by propagating the posterior distribution from the previous frame, using a constant velocity motion model. By modifying this transition model, velocities may be defined either in meters per frame, or (if the framerate is known) in meters per second. The likelihood $\mathbf{p}(\mathbf{i}|\mathbf{r}, \mathbf{v})$ is defined by the product of 2 terms based on appearance $\mathbf{p_a}(\mathbf{i}|\mathbf{r}, \mathbf{v})$ and depth $\mathbf{p_d}(\mathbf{i}|\mathbf{r}, \mathbf{v})$, assuming independence between the two modalities.

Appearance Sensors

When the observations include the outputs $\mathbf{A}_{1...M}$ of M appearance sensors (RGB or greyscale), the probability of a structure point at a given $\mathbf{r}$ and $\mathbf{v}$ can be estimated using the brightness constancy assumption, which states that the color of a point remains the same over time, and when viewed from any direction. If we assume the true color of a point is the average of the colors observed in each image (referred to as $\bar{I}_{r,v}$, defined in equation 2), then the squared error (divergence from brightness constancy) is equivalent to the variance of the projected color across all sensors at both the previous and current frame ($\mathbf{A}^{t-1}$ and $\mathbf{A}^t$ respectively). Using this cost function, a likelihood can be obtained as in equation 3, where $\Pi_m(\mathbf{r})$ is the projection function for camera m (returning the 2D pixel location vector), $\mathbf{r^t}$ is the current structure estimate and $\mathbf{r^{t-1}}$ is determined by $\mathbf{r} - \mathbf{v}$ (not by re-using the previous $\mathbf{r}$ estimate). This means the likelihood depends only on the current $\mathbf{r}$ and $\mathbf{v}$ values (i.e. it is memoryless). Also note that the employed cost function has a tighter peak and heavier tail (controlled by the parameter e_a), than the traditional Gaussian model of measurement noise. These characteristics have previously been found to more accurately reflect the statistics of motion estimation tasks [34]. In this paper e_a is set to 1, however training data could in principle be used to learn a more representative value.

$$\bar{I}_{r,v} = \sum_{\tau=t-1}^{t} \sum_{m=1}^{M} \frac{\mathbf{A}_m^\tau(\Pi_m(\mathbf{r}^\tau))}{2M} \tag{2}$$

$$\mathbf{p_a}(\mathbf{i}|\mathbf{r}, \mathbf{v}) = \frac{1}{1 + e_a \displaystyle\sum_{\tau=t-1}^{t} \sum_{m=1}^{M} \frac{\left(\mathbf{A}_m^\tau(\Pi_m(\mathbf{r}^\tau)) - \bar{I}_{r,v}\right)^2}{2M}} \tag{3}$$

Depth Sensors

When depth information is present, the task is simplified, as most ambiguity in $\mathbf{r}$ is removed, and depth sensors can contribute to the likelihood calculation. Given observations $\mathbf{D}_{1...L}$ from a collection of L depth sensors, the likelihood of a true structural point at $\mathbf{r}^t$ should fit well with the back-projections of all depth sensors. Similarly $\mathbf{r}^{t-1}$ should match the back-projection of the previous depth observations. To quantify this, the average square distance between the position $\mathbf{r}$, and each sensors back-projection, is calculated as in equation 5, where Ψ_l is the projection function for depth sensor l, and Ψ'_l is the corresponding back-projection function. This back-projection takes the pixel position and depth value (defined as the 3 dimensional vector $\tilde{\mathbf{r}}$), obtained from the sensor, and returns the 3D vector $\mathbf{r}$ of the corresponding structure point.

$$\tilde{\mathbf{r}}_l^\tau = \{\Psi_l(\mathbf{r}^\tau), \mathbf{D}_l^\tau(\Psi_l(\mathbf{r}^\tau))\} \tag{4}$$

$$\mathbf{p_d}(\mathbf{i}|\mathbf{r}, \mathbf{v}) = \frac{1}{1 + e_d \displaystyle\sum_{\tau=t-1}^{t} \sum_{l=1}^{L} \frac{(\Psi'_l(\tilde{\mathbf{r}}_l^\tau) - \mathbf{r}^\tau)^2}{2L}} \tag{5}$$

Note that the e_a and e_d parameters may be modified to control the relative contribution of the appearance and depth information. Also note that if either M or L are zero (only one type of sensor is present) the relevant term is reduced to 1, and $\mathbf{p}(\mathbf{i}|\mathbf{r}, \mathbf{v})$ depends entirely on the remaining sensor modality (assuming that the sum of an empty set is defined as zero). Thus the approach is generalizable to any combination of inputs.

4 Scene Particles

Evaluating $\mathbf{p}(\mathbf{r}, \mathbf{v}|\mathbf{i})$ densely across even a quantized 6D space is intractable without a severe loss of fidelity. This is due to the large number of samples required. As an example, even a simple setup using only 2 cameras of 640 by 480 resolution, leads to millions of possible structure points, each with an equivalent number of possible velocities. Instead, a discrete subset of weighted samples (termed "Scene Particles") is maintained. Similar to particle filtering approaches [35], [36], [37], this allows us to approximate a continuous probabilistic system, while remaining computationally feasible.

Each of the N Scene Particles ($\mathbf{P_{0\dots N}}$) is comprised of a 3D spatial location and 3D velocity (i.e. a pair of $\mathbf{r}$ and $\mathbf{v}$ such that $\mathbf{P_n} \in \mathbb{R}^6 = \{\mathbf{r}_n, \mathbf{v}_n\}$). Additionally, each Scene Particle and has a weight w_n, obtained by analyzing $\mathbf{p}(\mathbf{i}|\mathbf{r}, \mathbf{v})$ at the Scene Particle.

$$w_n = \mathbf{p}(\mathbf{i}|\mathbf{r}_n, \mathbf{v}_n) = \mathbf{p}_a(\mathbf{i}|\mathbf{r}_n, \mathbf{v}_n)\mathbf{p}_d(\mathbf{i}|\mathbf{r}_n, \mathbf{v}_n) \quad (6)$$

As new observations are obtained, the Scene Particles from the previous frame are propagated using a constant velocity motion model. This provides a sampled approximation of the prior probability $\mathbf{p}(\mathbf{r}, \mathbf{v})$, allowing the principled propagation of information between frames. This is frequently absent in scene flow estimation, due to long processing times which place greater emphasis on single frame accuracy. The density of Scene Particles from the prior, combined with reweighting by the likelihood, leads to the approximation of the posterior. The system is initialized at the first frame with a randomly generated, uniformly weighted, Scene Particle cloud (i.e. a uniform prior).

As in particle filtering systems, a resampling scheme is employed to concentrate hypotheses into promising regions of the probability distribution. The residual resampling scheme [38] is employed, where a new Scene Particle population $\mathbf{\Theta^{t+1}}$ is sampled from the previous population $\mathbf{\Theta^t}$, with the probability of choosing Scene Particle $\mathbf{P_n}$ equal to w_n.

When observations from depth sensors are present, an additional preprocessing stage is performed on the Scene Particles, to represent the reduction in spatial ambiguity. The projections of each Scene Particle $\mathbf{P_n}$ are calculated in all depth sensors $\mathbf{D}_{1\dots L}$. The subset of sensors with a valid depth estimate at the projected location is extracted, and one is randomly selected, to provide a new spatial location $\mathbf{r}$ for $\mathbf{P_n}$. In the case where some sensors observe structure which is occluded in other sensors, the random selection process ensures the resulting particle population reflects the proportions of visibility.

4.1 Iterative Estimation

As each new set of observations $\mathbf{i}$ is obtained, the likelihood $\mathbf{p}(\mathbf{i}|\mathbf{r}, \mathbf{v})$ is iteratively estimated over s scales $\mathbf{i_{0\dots s}}$, from coarse to fine [31], [39], [40]. This aids Scene Particles in avoiding small local maxima from incorrect correspondences. After each scale iteration, resampling of the population is performed, with Gaussian diffusion. To further aid convergence, an additional iterative estimation is performed at each scale. For each iteration of this inner loop, the level of Gaussian diffusion is halved. This allows particles to begin with a more exploratory behavior, and then to transition to a more precise convergence.

4.2 Overconvergence

In standard particle filtering algorithms, each hypothesis relates to a complete solution to the task. However, in the Scene Particle algorithm we wish to obtain a large collection of high probability samples. This mismatch leads to problems over time, as it is well known that repeated iterations will eventually cause particle filters to converge to a single mode [41]. To reduce this effect, the resampling scheme is modified using a technique we refer to as Ray Resampling.

Before resampling, the particle population is partitioned into groups based on the optical rays the particles lie along. Standard residual resampling is then applied to each partition, in order to fill equally sized subsets of the following population. More formally, for a system with R rays, $\frac{|\mathbf{\Theta}|}{R}$ new particles are created by each partition. First, each particle $\mathbf{P_n}$ in the partition is added to the following population $\bar{w}_n \frac{|\mathbf{\Theta}|}{R}$ times, where $\bar{w}_n$ is the particles weight, normalized by the total weight within the partition. Secondly any samples remaining to be drawn (due to rounding errors) are chosen randomly, based on the residual weights of the particles within the partition [38].

This process is applied simultaneously across all viewpoints, meaning every input particle is part of multiple resampling groups. Despite this, the resulting population has the same size as the original, as the number of new samples per partition is the same regardless of how many particles it contains. This approach to resampling ensures that every area of the scene is represented by an equal number of Scene Particles, guaranteeing coverage. Additionally, the resultant Scene Particle cloud has the valuable property of being dense in all viewpoints, as opposed to most scene flow and Stereo Reconstruction schemes, which are dense in a single, arbitrarily selected, reference view. Ray Resampling could also be seen as using a separate particle filter for each ray (similar to the per-pixel Kalman filters of [19]) while allowing hypotheses to exist within multiple particle filters, and to transition between them across frames.

After the newly resampled population is generated, the particles are uniformly weighted, such that the prior is encoded entirely by the density of samples. This is in contrast to the formulation previously presented [20], where the prior was present in both the sampling density, and the sample weights. This lead to greatly accelerated convergence rates, and prompted an update rule breaking with the standard Bayesian formulation, which is not necessary here.

Occasionally no Scene Particles will fall along an optical ray, in this case, Scene Particles are randomly selected from those falling on neighboring rays, reflecting the surrounding distributions.

4.3 Occlusion

When employing multiple viewpoints, particularly with very large baselines and differences in orientation, large areas of the scene may be visible to only a subset of sensors. Equations 3 and 5 make no allowances for occlusion, favoring only estimates which are consistent across all viewpoints.

An estimate of how likely a region is to suffer from occlusion in a given sensor m at frame t, can be obtained from the current prior probability $\mathbf{p}(\mathbf{r}, \mathbf{v})$, with a high occlusion chance identified if the viewing ray between the sensor and the region in question intersects a point of high prior probability. To quantify this, equation 7 details the calculation of the visibility probability distribution $\mathbf{O}_{m,t}$ where u is the viewing ray of m intersecting $\mathbf{r}$. As occlusion does not depend on the instantaneous velocity, $\mathbf{v}$ is marginalized out of the prior to produce $\mathbf{p}^t(\mathbf{r})$. Again this calculation is performed solely using the current system state, thus $\mathbf{p}^{t-1}(\mathbf{r})$ is estimated via $\mathbf{r} - \mathbf{v}$ before marginalization (not by re-using the prior from the previous frame).

$$\mathbf{O}_{m,t}(\mathbf{r}) = \frac{\mathbf{p}^t(\mathbf{r})}{\int_0^r \mathbf{p}^t(u)\mathrm{d}u} \tag{7}$$

As the integral along u to $\mathbf{r}$ includes $\mathbf{p}(\mathbf{r})$, $\mathbf{O}_{m,t}$ is limited to 1 when there is no probability along the ray prior to $\mathbf{r}$. Thus it can be seen that $0 \leq \mathbf{O}_{m,t} \leq 1$. Figure 2 illustrates the behavior of this distribution in both the continuous and discrete cases. Intuitively, the visibility probability relates to the weight of a point, in comparison to the cumulative weight of all points before it. A separate visibility probability map can be created for each viewpoint, and may be used in other tasks. Note that unlike many previous approaches to motion estimation [29], [42], [40], the evaluation of a points validity and it's occlusion status are entirely separate. Rather than determining occlusions simply by the lack of observational consistency (which is employed in the likelihood function, to determine valid motions), the presented formulation requires occlusions to be justified by the previously estimated structure and motion fields, i.e. points are not labeled as occluded if there is nothing to do the occluding.

To utilize this information and improve scene flow estimation, the likelihood equation 3 is modified, to give increased importance to consistency between viewpoints which are unlikely to be occluded, as shown in equation 8, where $\bar{\mathbf{O}}$ is the total visibility probability across all observations.

$$\mathbf{p_a}(\mathbf{i}|\mathbf{r},\mathbf{v}) = \frac{1}{1+e_a \sum_{\tau=t-1}^{t} \sum_{m=0}^{M} \frac{\mathbf{O}_{m,\tau}(\mathbf{r})\left(\mathbf{A}_m^{\tau}(\Pi_m(\mathbf{r}^{\tau}))\text{-}\bar{I}_{r,v}\right)^2}{2M\bar{\mathbf{O}}}} \tag{8}$$

$$\bar{\mathbf{O}} = \sum_{\tau=t-1}^{t} \sum_{m=0}^{M} \mathbf{O}_{m,\tau}(\mathbf{r}) \tag{9}$$

A similar modification is applied to the structural likelihood equation 5. Both the scene flow estimate, and visibility distributions converge during iterative estimation. However, the system is susceptible to the degenerate case, where points are visible in only a single view, and data consistency is examined only within that view. To prevent this, an additional factor could be introduced based on the sum of the squared visibility probabilities, which would favor points with uniform visibility scores. In this paper, hard constraints are instead used, relating to the constraints required to determine scene flows. Hypotheses are eliminated if they do not exhibit greater than average visibility in at least 3 images (including at least one from both t and $t-1$).

4.4 Motion Smoothing

The maintenance of multiple hypotheses in the Scene Particles algorithm allows information from past frames to aid the disambiguation of motion, rather than relying on smoothness constraints, which produce artifacts at discontinuities. However, the assumption of smoothness in the motion field is still valid in many cases, and so we explore a number of techniques for exploiting this information, without compromising the benefits of the algorithm.

4.4.1 2D Post Filtering

The simplest approach is to follow the suggestions of previous authors [43], [42], and apply smoothing to the image plane projection of the scene. This is referred to as I_p which is an image comprising of 4 channels (v_x, v_y, v_z and z) and is created based on the projections of the Scene Particle cloud onto camera 0. Each pixel in I_p is filled by taking the weighted average of the v_x, v_y, v_z and z values of all Scene Particles along the associated ray, using their weights w_n. It is important to note that filtering I_p does not affect the Scene Particle population, only the estimated motion field derived from them. As such, errors occurring from smoothing are not accumulated over time.

In order to minimize oversmoothing of discontinuities, both in the structure, and the motion estimates, a bilateral filtering technique is used. Bilateral filtering is biologically inspired and reflects the way human vision operates. In essence, the bilateral filter performs Gaussian smoothing within a region, with an additional reweighting of contributions based on value dissimilarity. The filtered values $\hat{I}_p(j,k)$ at pixel j,k can be produced from a q by q image patch (represented by the set Ω_2 of 2D offset vectors), with each motion channel (v_x, v_y, v_z) being processed independently. Equation 10 defines the creation of the smoothed v_x value (the channel of I_p is represented in superscript), with related definitions for the remaining channels. In addition, $g(x,\sigma)$ refers to a zero mean

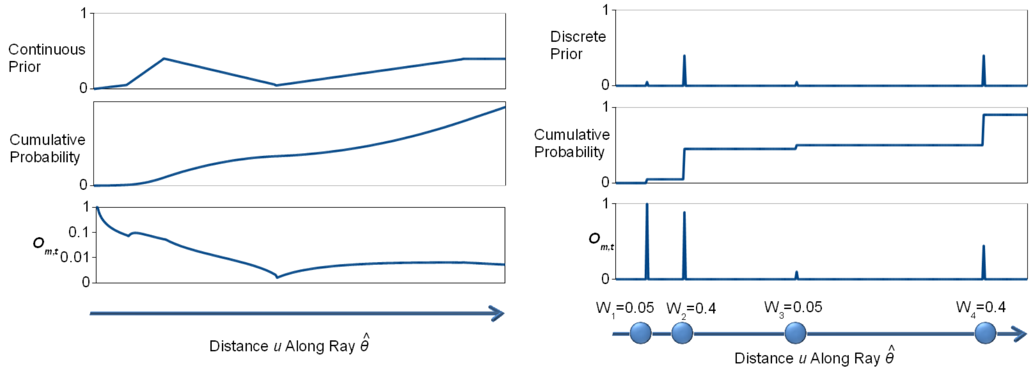


Fig. 2: Visibility Probability calculations as discussed in section 4.3, in the case of continuous and discrete priors. The discrete example is composed of 4 Scene Particles of varying weights. Note that the visibility probability for particle 2 is high, as its weight is much greater than that of particles in front of it.

Gaussian function with standard deviation of σ, evaluated at x.

$$\hat{I}_p^{vx}(j,k) = \sum_{\omega\in\Omega_2(j,k)} I_p^{vx}(\omega) g\left(|\omega|,\sigma_{s2}\right) g\left(I_p^{vx}(j,k)\text{-}I_p^{vx}(\omega),\sigma_{v2}\right) \tag{10}$$

$$\Omega_2(j,k) = \{(j',k') : |j-j'| < q \text{ and } |k-k'| < q\} \tag{11}$$

Using bilateral filtering significantly reduces smoothing artifacts when compared to a simple Gaussian weighted neighborhood. The value of σ_{v2} relates to the size of discontinuities expected in the estimated values, discontinuities larger than 2 standard deviations will suffer little oversmoothing. This is useful for preserving boundaries, which is important for object segmentation tasks, however, fine details smaller than 1 standard deviation will likely be unrecoverable.

4.4.2 3D Post Filtering

More advanced than filtering the 2D projection I_p, is to filter the 3D scene directly, via convolution with a 3D kernel. In contrast to the image plane formulation, this allows smoothness constraints to be confined to a small region of space, maintaining finer details and preventing Scene Particles separated in z from influencing each other. Further, by removing projective distortions, assumptions on motion and structural smoothness are more valid. The smoothed Scene Particle $\hat{\mathbf{P}}_\mathbf{n} = \{\mathbf{r},\hat{\mathbf{v}}\}$ can be constructed from the unsmoothed Scene Particle $\mathbf{P}_\mathbf{n} = \{\mathbf{r},\mathbf{v}\}$ and the set of neighboring Scene Particles Ω_3 (defined as the particles within 2 spatial standard deviations σ_{s3}), by calculating each component of the smoothed motion $\hat{\mathbf{v}}$ independently, according to equation 12 while leaving $\mathbf{r}$ unchanged.

$$\hat{\mathbf{P}}_n^{vx} = \sum_{P_q\in\Omega_3(\mathbf{P}_n)} \mathbf{P}_n^{vx} g\left(|\mathbf{r}_n\text{-}\mathbf{r}_q|,\sigma_{s3}\right) g\left(\mathbf{P}_n^{vx}\text{-}\mathbf{P}_q^{vx},\sigma_{v3}\right) \tag{12}$$

$$\Omega_3(\mathbf{P}_n) = \{\mathbf{P}_q : |\mathbf{r}_n - \mathbf{r}_q| < 3\sigma_{s3}\} \tag{13}$$

5 Algorithm Summary

Pseudocode for the Scene Particles algorithm operating on a single frame, is provided in algorithm 1. The majority of the computation is performed between lines 1 and 16, within the nested scale and diffusion loops mentioned in section 4.1. At each iteration, the visibility probabilities from section 4.3 are calculated for each particle, followed by the particles likelihood. Ray resampling and diffusion is also performed at each iteration. When iterative estimation is complete, particles are assigned their final likelihoods, the output image I_p is generated for evaluation purposes, and the particle population is propagated via a constant velocity motion model to produce the prior for the following frame.

6 Scene Flow Estimation Results

For conciseness, results discussed here are quantitative in nature. A range of additional datasets and results, including long sequences including both appearance and depth inputs, are available online [1] and as supplementary material. A small number example frames are presented in figure 3.

All experiments were performed using a C++ implementation, available from the above link. Unless otherwise stated, 20 Scene Particles were used per ray, and with particle filter bounds equal to 50% of the maximum visible velocity. In addition, the standard deviation of the Gaussian diffusion during resampling was equal to 3% of the maximum visible velocity, reducing by 30% at each inner loop iteration, while the outer loop made use of 6 image scales, halving at each stage. Note that these parameters have been related to the maximum velocity to aid comparisons between scenes, however this is not necessary in practice.

The quantitative results in this section are estimated on two types of data. The first is the Middlebury datasets [44], which were originally developed as a

[1] personal.ee.surrey.ac.uk/Personal/S.Hadfield/sceneparticles

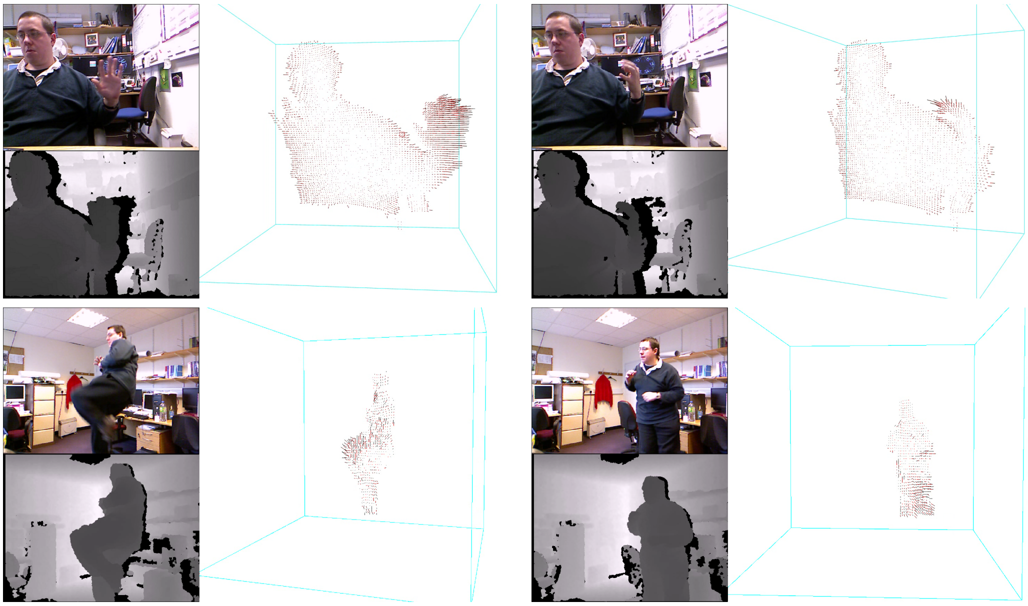


Fig. 3: Example frames of scene flow estimation with a Kinect. For each frame, the appearance and depth inputs are shown, alongside the 3D flow field. Flow vectors travel from the red to the purple vertices. The top row relates to the hands sequence. The bottom row relates to the kicks sequence.

Algorithm 1 Scene Particles

```
for scales $s = 0$ to $S$ do
  for diffusion level $\delta = 0$ to $\Delta$ do
    for all particles $\mathbf{P}_n$ in $\mathbf{\Theta^t}$ do
      if Occlusion aware then
        for m $= 0$ to $M$ and t $= 0$ to $1$ do
          calculate occlusion $\mathbf{O}_{m,t}(\mathbf{r}_n) = \frac{\mathbf{p}^t(\mathbf{r}_n)}{\int_0^{r_n} \mathbf{p}^t(u)\mathrm{d}u}$
        end for
      end if
      calculate weight $w_n = \mathbf{p}_a(\mathbf{i}|\mathbf{r}_n, \mathbf{v}_n)\mathbf{p}_d(\mathbf{i}|\mathbf{r}_n, \mathbf{v}_n)$
    end for
    for ray $= 0$ to $R$ do
      for distance along ray from $0$ to $\frac{|\Theta^t|}{R}$ do
        $\Theta^{t+1} \cup \mathbf{P}_n$ where $\mathbf{P}_n \sim \mathbf{p}(\mathbf{r}, \mathbf{v}|\mathbf{i})$ and $\mathbf{P}_n \in$ ray
      end for
    end for
    for all $\mathbf{P}_n$ in ray population $\mathbf{\Theta^t}$ do
      $\mathbf{P}_n = \mathbf{P}_n + \delta\xi,$ where $\xi \sim g(0, \sigma)$
    end for
  end for
end for
for all particles $\mathbf{P}_n$ in ray population $\mathbf{\Theta^t}$ do
  $w_n = \mathbf{p}_a(\mathbf{i}_0|\mathbf{r}_n, \mathbf{v}_n)\mathbf{p}_d(\mathbf{i}_0|\mathbf{r}_n, \mathbf{v}_n)$
end for
for all particles $\mathbf{P}_n$ in ray population $\mathbf{\Theta^t}$ do
  $\mathbf{r}_n = \mathbf{r}_n + \mathbf{v}_n$
end for
```

benchmark for stereo reconstruction algorithms. This demonstrates the performance of the algorithm on non-synthetic data with complex scenes. It is useful to evaluate using such data, as Vaudrey *et al.* [45] found that performance on synthetic data often leads to fundamentally different behavior than in real applications. The second set of data used is the multiview rotating sphere of Basha *et al.*[33]. This dataset consists of a structurally simple synthetic scene, but involves a much more complex, discontinuous, motion field.

6.1 Isolated Estimation

Most current scene flow estimation techniques take many hours to process each frame, as such results are generally analyzed on a single frame. To achieve this a number of 2 frame sequences are constructed to simulate moving cameras. For the Middlebury datasets, taking images $\{I_1, I_3, I_5, I_7\}$ as the first frames of 4 sequences, and images $\{I_2, I_4, I_6, I_8\}$ as the following frames, allows us to simulate 4 cameras, each translating by $\Delta\mathbf{x}$. This is equivalent to a set of 4 stationary cameras, viewing a scene in which every point translates by $-\Delta\mathbf{x}$. Although the motion field is simple when viewed in 3D, it's projection onto the images is very complex. None of the techniques listed in the following sections make use of any prior knowledge about this scene, including the rigidity of the objects.

To convert the Scene Particle cloud into a form comparable to that of other techniques, I_p is analyzed. As mentioned in section 4.4, the motion at each pixel is taken as the weighted average of the motion, of all Scene Particles along the relevant optical ray. Where needed, the output of depth sensors is simulated by projecting the ground truth structure to a number of equally spaced image planes. Following Basha *et al.* [40], errors for the Middlebury datasets are analyzed in the image plane, using the average RMS error of the optical flow (RMS-OF), disparity (RMS-Z) and disparity flow (RMS-Vz) which are measured in pixels, in conjunction with the absolute angular error (AAE) which measures directional accuracy in degrees. Error measures are averaged over all pixels in I_p (unlike previous authors, occlusion regions were not excluded from the analysis). Also to allow comparison to Basha *et al.* errors for the Sphere sequence are analyzed in the 3D space, using the normalized RMS errors in 3D position (%NRMS-P), 3D velocity magnitude (%NRMS-V), and 3D velocity angular error (AAE-V) [40]. The normalized errors are equivalent to standard RMS errors, but presented as a percentage of the range of ground truth motions (i.e. divided by the difference between the minimum and maximum ground truth value, then multiplied by 100 as proposed in [40]). Error measures in the 3D space are presented this way, because the 3D reconstruction is accurate only up to an arbitrary scale factor, e.g. two datasets generated with different focal lengths, would have incompatible RMS error scores before normalization.

6.1.1 Multi-Modal Scene Flow

No current scene flow estimation algorithm is capable of utilizing observations from both appearance and depth sensors. For comparison purposes, a state of the art optical flow algorithm [46] was applied to the appearance information, and the results are combined with the depth data to infer 3D motion. This technique is referred to as OFD. For the experiments labeled GT in table 1, depth observations were produced using the ground truth depth. For all other experiments, depth observations were produced using the semi global matching stereo algorithm of Hirschmuller [47]. This produces depth maps of significantly lower quality than observations from true depth sensors, such as those shown in figure 3. However, the use of estimated depth maps is similar to the stereo initialization utilized by competing approaches [40].

Unsurprisingly, the out of plane motion accuracy for the Scene Particles algorithm is greater than that of the OFD approach. However, it is interesting to note that despite OFD utilizing a dedicated optical flow algorithm, the motion magnitude error within the image plane is also worse. This implies that the incorporation of depth data at an earlier stage, allows more accurate flow estimates, even in 2D. The OFD algorithm provides greater scene coverage than standard Scene Particles, but still cannot achieve the 100% coverage of Ray Resampled Scene Particles.

The OFD technique displays the greatest robustness to the quality of the depth input, which does not affect the quality of the in plane and directional errors. In addition, the reduction in out of plane accuracy is surprisingly low. This is likely because regions in which stereo reconstruction fails frequently correspond to regions where the optical flow fails (and where errors cannot be evaluated).

It would be reasonable to assume that the over-convergence of the standard Scene Particle algorithm, would lead to the accumulation of particles around regions of high accuracy, causing reduced coverage and correspondingly increased accuracy. However this is not always the case, in fact the performance of the standard resampled and ray resampled approaches are roughly equivalent. This implies that, within the subset of local maxima of $\mathbf{p}(\mathbf{i}|\mathbf{r},\mathbf{v})$, higher probabilities do not necessarily relate to reduced ambiguity.

6.1.2 Appearance Only Scene Flow

From figure 4 it can be seen that, unlike traditional motion estimation techniques, the Scene Particles algorithm does not suffer from reduced accuracy at discontinuities. Instead, the independence of the scene particles leads to low levels of error, spread uniformly across the scene. This is a useful property as, for many applications such as robot navigation and action recognition, object boundaries prove particularly salient.

These results are also analyzed over a single frame. As a result, there is no propagation of information through time, and much of the improvements in accuracy can likely be attributed to this lack of over-smoothing artifacts. Observing the performance in table 2, it can be seen that the use of 3 additional appearance sensors compensates for the loss of the depth sensor input, with motion accuracy showing moderate improvement over the results in table 1. Unsurprisingly, the use of depth observations still greatly improves structural estimation accuracy (RMS-Z).

The Scene Particle algorithm consistently estimates motion magnitude, more accurately than previous approaches, both within the image plane, and perpendicular to it. However, directional estimation accuracy is slightly lower than existing techniques. This is due to the stochastic nature of the Scene Particles algorithm, the motion fields always suffer from a low level of noise (as seen in figure 4). In terms of motion magnitude, this noise is generally insignificant, however in areas of low motion (such as background regions) a small change in absolute motion leads to a large shift in direction. The probabilistic occlusion approach seems to somewhat mitigate this, while also improving motion magnitude accuracy.

To make runtime comparisons fairer, the speeds listed are for sequential computation in a single thread, not exploiting the possibility for massive parallelization provided by the Scene Particle's independence. Exact computation time was not provided by Basha *et al.* however, it is stated to be of the same order as that of Huguet *et al.* The standard Scene Particles approach operates around 100 times faster than previous approaches, with the occlusion aware version increasing runtimes by around 30%.

6.2 Information Propagation

As the Scene Particle algorithm is several orders of magnitude faster than previous techniques, it is useful

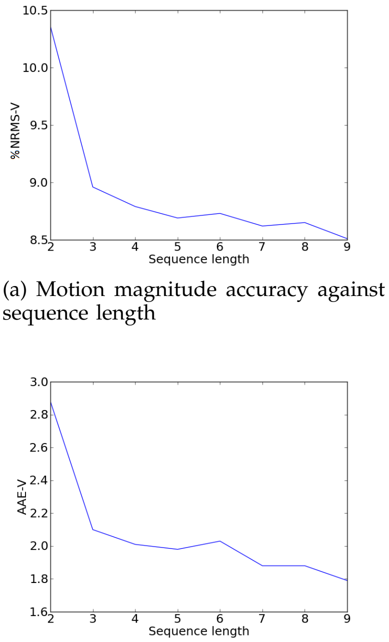
Algorithm	Dataset	App Sens.	Depth Sens.	RMS-OF	RMS-Vz	AAE	Coverage
Scene Particles	Cones	1	1	0.70	0.02	1.58	28.59
Scene Particles + RR	Cones	1	1 (GT)	**0.59**	**0.01**	1.61	**100.00**
Scene Particles + RR	Cones	1	1	0.60	0.02	1.64	**100.00**
OFD [46]	Cones	1	1 (GT)	2.30	1.57	**0.52**	86.79
OFD [46]	Cones	1	1	2.30	1.60	**0.52**	86.79
Scene Particles	Teddy	1	1	**0.50**	**0.01**	1.63	20.72
Scene Particles + RR	Teddy	1	1 (GT)	0.52	**0.01**	1.36	**100.00**
Scene Particles + RR	Teddy	1	1	0.60	**0.01**	1.35	**100.00**
OFD [46]	Teddy	1	1 (GT)	2.11	0.69	**0.43**	91.50
OFD [46]	Teddy	1	1	2.11	0.70	**0.43**	91.50
Scene Particles	Venus	1	1	**0.53**	**0.00**	2.58	14.51
Scene Particles + RR	Venus	1	1 (GT)	0.72	**0.00**	2.62	**100.00**
Scene Particles + RR	Venus	1	1	0.84	**0.00**	2.86	**100.00**
OFD [46]	Venus	1	1 (GT)	1.16	0.28	**0.61**	97.45
OFD [46]	Venus	1	1	1.16	0.30	**0.61**	97.45

Algorithm	Dataset	App Sens.	Depth Sens.	%NRMS-V	AAE-V	Coverage
Scene Particles	Sphere	1	1	12.65	1.95	3.06
Scene Particles + RR	Sphere	1	1 (GT)	**10.26**	**2.68**	**100.00**
Scene Particles + RR	Sphere	1	1	10.36	2.88	**100.00**
OFD [46]	Sphere	1	1 (GT)	12.42	5.01	91.25
OFD [46]	Sphere	1	1	12.42	5.12	91.25

TABLE 1: Results of scene flow estimation in a combined depth and appearance sensor system. Scene Particles with and without ray resampling (RR) are compared to the combined optical flow [46] and depth estimation scheme. Discussion is contained in section 6.1.1.

to analyze the performance of the algorithm on longer sequences, with more relevance to practical applications. The propagation of information over time is a unique aspect of the Scene Particles algorithm, which makes it particularly suited to this situation.

Figure 5 shows the performance of the Scene Particles algorithm when run on sequences of varying lengths. Adding a single additional frame to the sequences, causes a considerable reduction in both directional and motion magnitude errors. However, the improvement from subsequent additional frames, is less significant (although still nonzero). This is to be expected, as groups of motion hypotheses rarely remain ambiguous across more than two sets of observations. The observed lower limit on the performance, is likely due to regions emerging from occlusion, for which the prior information does not help, and the use of a constant velocity motion model when creating the prior distribution.



(a) Motion magnitude accuracy against sequence length

(b) Directional accuracy against sequence length

Fig. 5: Analysis of error measurements on the sphere dataset, when simulating sequences of various lengths. Experiments performed with a single appearance and depth sensor.

6.3 Smoothing

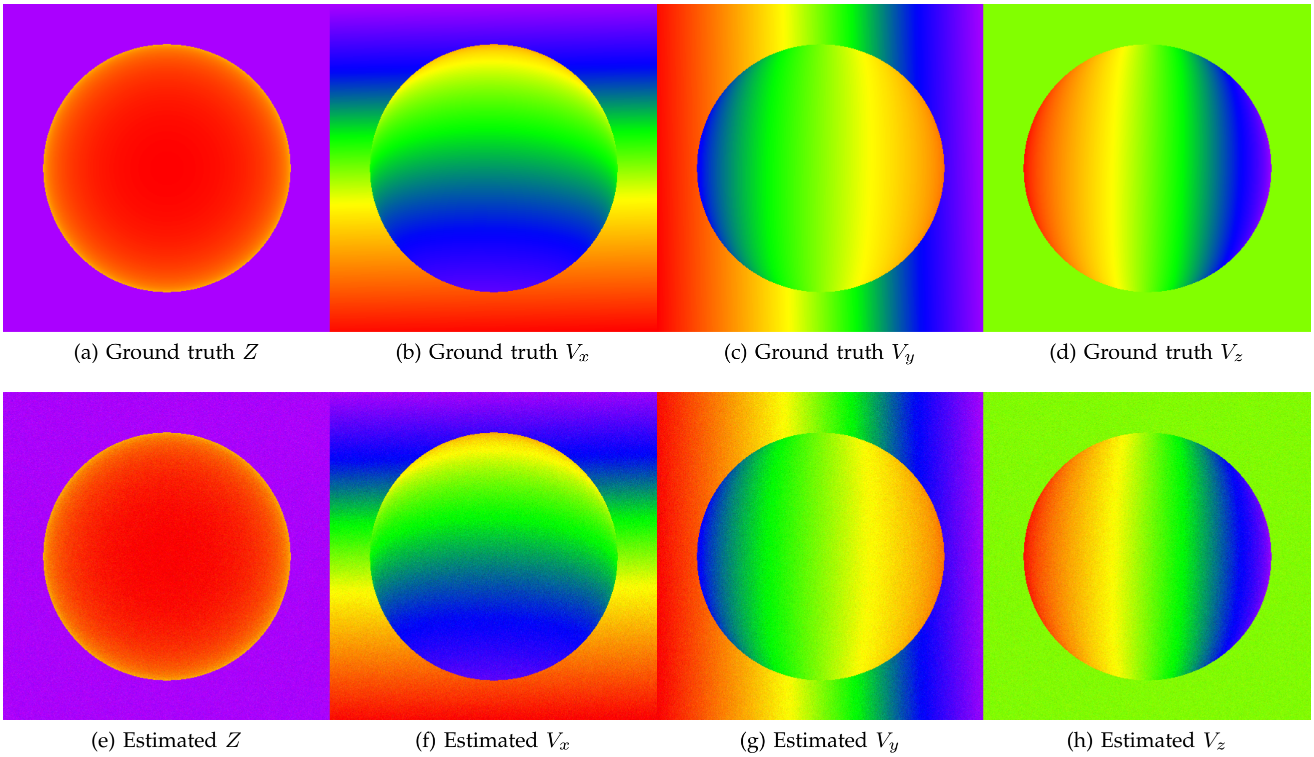
A number of approaches to incorporating smoothness constraints into scene flow estimation are possible. The techniques used for comparison in the previous section rely on regularization terms in the cost function during optimization. Another common approach is to define the matching function to compare patches between viewpoints and times, rather than a single pixel. In table 3, we analyze the effect this has on the Scene Particle algorithm, in addition to the smoothing post-process techniques discussed in section 4.4.

The use of patch based matching and 2D post filtering show little improvement in terms of motion magnitude accuracy. However, by applying 3D post filtering, a significant increases both in magnitude and directional accuracy can be obtained, at the cost of doubled computation time.

Algorithm	Dataset	App Sens.	Depth Sens.	RMS-OF	RMS-Vz	RMS-Z	AAE	Time/Frame
Scene Particles (multiview)	Cones	4	0	0.27	0.02	2.44	1.74	344 secs
Scene Particles (multiview)	Cones	4	1	0.25	0.01	**1.42**	2.22	327 secs
Scene Particles + (Occ.)	Cones	4	0	**0.22**	0.02	2.40	1.32	418 secs
Basha *et al.*[40]	Cones	2	0	0.58	0.01	2.48	0.39	-
Basha *et al.*[40]	Cones	4	0	0.25	**0.00**	2.36	**0.12**	-
Huguet *et al.*[39]	Cones	2	0	1.10	3.13	2.11	0.69	5 hours
Scene Particles (multiview)	Teddy	4	0	0.18	0.01	1.40	1.19	401 secs
Scene Particles (multiview)	Teddy	4	1	0.17	0.01	**0.77**	2.14	348 secs
Scene Particles (Occ.)	Teddy	4	0	**0.13**	0.01	1.48	1.16	893 secs
Basha *et al.*[40]	Teddy	2	0	0.57	0.03	2.83	1.01	-
Basha *et al.*[40]	Teddy	4	0	0.51	**0.00**	2.47	**0.22**	-
Huguet *et al.*[39]	Teddy	2	0	1.25	4.66	2.27	0.51	5 hours
Scene Particles (multiview)	Venus	4	0	**0.07**	**0.00**	0.73	2.05	312 secs
Scene Particles (multiview)	Venus	4	1	0.09	**0.00**	**0.40**	2.32	337 secs
Scene Particles (Occ.)	Venus	4	0	**0.07**	**0.00**	0.73	1.05	423 secs
Basha *et al.*[40]	Venus	2	0	0.16	**0.00**	1.06	1.58	-
Basha *et al.*[40]	Venus	4	0	0.13	**0.00**	0.90	1.09	-
Huguet *et al.*[39]	Venus	2	0	0.31	0.51	0.97	**0.98**	5 hours

Algorithm	Dataset	App Sens.	Depth Sens.	%NRMS-V	%NRMS-P	AAE-V	Time/Frame
Scene Particles (multiview)	Sphere	5	0	9.41	6.36	3.04	1053 secs
Scene Particles (multiview)	Sphere	5	1	8.64	**2.97**	3.16	990 secs
Scene Particles + (Occ.)	Sphere	5	0	**8.39**	5.97	**3.02**	1818 secs
Basha *et al.*[40]	Sphere	5	0	9.71	4.39	3.39	-

TABLE 2: Results of scene flow estimation for a multi-view, appearance only setup. Scene Particles with and without occlusion awareness are compared to Basha *et al.* [40] and Huguet *et al.* [39]. Discussion is contained in section 6.1.2.



(a) Ground truth Z (b) Ground truth V_x (c) Ground truth V_y (d) Ground truth V_z

(e) Estimated Z (f) Estimated V_x (g) Estimated V_y (h) Estimated V_z

Fig. 4: Ground truth (top) and estimated (bottom) images for the scene structure and motion in the Sphere dataset, using the Scene Particle algorithm with 5 appearance sensors only.

Algorithm	Dataset	RMS-OF	RMS-Vz	RMS-Z	AAE	Time/Frame
Scene Particles	Cones	0.22	0.02	**2.40**	1.32	418 secs
Scene Particles (Patch Cost)	Cones	0.37	0.02	3.37	**0.69**	13681 secs
Scene Particles (2D Post Filter)	Cones	0.21	**0.01**	2.39	1.32	505 secs
Scene Particles (3D Post Filter)	Cones	**0.19**	**0.01**	**2.40**	1.21	824 secs
Scene Particles	Teddy	0.13	0.01	**1.48**	1.16	493 secs
Scene Particles (Patch Cost)	Teddy	0.16	0.01	3.28	**0.72**	9025 secs
Scene Particles (2D Post Filter)	Teddy	0.13	0.01	**1.48**	1.17	333 secs
Scene Particles (3D Post Filter)	Teddy	**0.11**	**0.00**	**1.48**	1.14	890 secs
Scene Particles	Venus	0.07	**0.00**	**0.73**	1.05	423 secs
Scene Particles (Patch Cost)	Venus	0.82	**0.00**	2.13	**0.33**	10114 secs
Scene Particles (2D Post Filter)	Venus	0.08	**0.00**	**0.73**	1.15	447 secs
Scene Particles (3D Post Filter)	Venus	**0.05**	**0.00**	**0.73**	0.93	1022 secs

Algorithm	Dataset	%NRMS-V	%NRMS-P	AAE-V	Time/Frame
Scene Particles	Sphere	8.39	**5.97**	3.02	1818 secs
Scene Particles (Patch Cost)	Sphere	7.99	12.30	**2.06**	21332 secs
Scene Particles (2D Post Filter)	Sphere	8.12	**5.97**	2.91	1992 secs
Scene Particles (3D Post Filter)	Sphere	**5.57**	**5.97**	3.04	3309 secs

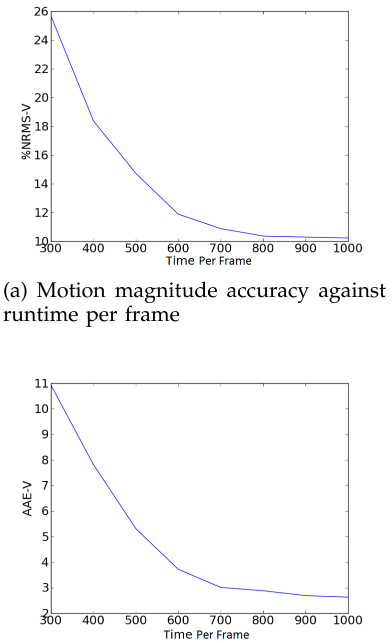
TABLE 3: The performance of the Scene Particles algorithm when incorporating smoothness constraints in a variety of ways. Tests are performed with occlusion aware Scene Particles, using 4 appearance sensors. Discussion is contained in section 6.3.

As the bilateral post filtering techniques are applied only to the motion field, structural performance does not change. However, using patch based matching significantly reduces structural estimation accuracy, to levels comparable with previous approaches in table 2. In addition, directional performance is improved, bringing it closer to the levels of existing techniques. This is an interesting finding, and implies that there are deep similarities between a global regularization scheme, and local patch based smoothing. Further, it demonstrates that the implementation of smoothness constraints is a limiting factor of current motion estimation performance.

6.4 Sampling Sparsity

A useful property of the Scene Particle algorithm is that varying the size of the Scene Particle population enables a tradeoff of accuracy and computation time. More hypotheses requires more time sampling the probability distribution, but increases the chances of locating the maxima of the distribution. Figure 6 shows the relationship between runtime and accuracy, obtained by varying the number of hypotheses per ray between 1 and 40.

As the number of hypotheses per ray increases, all error measurements exhibit exponential decay. At around 10 minutes per frame, directional accuracy plateaus at just under 3 degrees, highlighting the fundamental limit of stochastic estimation when analyzed using this metric. Motion magnitude errors saturates at a larger number of hypotheses, relating to runtimes of around 15 minutes per frame. Even at this speed, the scene particles algorithm is still an order of magnitude faster than competing approaches.



(a) Motion magnitude accuracy against runtime per frame

(b) Directional accuracy against runtime per frame

Fig. 6: Performance on sphere dataset against runtime, obtained by varying the number of particles. Results using a single appearance and depth sensor.

6.5 Robustness

To analyze the robustness of the approach, the input appearance and depth images were corrupted with varying degrees of noise, with results shown in figure 7. When testing Gaussian noise, every pixel value was corrupted by a Gaussian distributed value, performance was then analyzed as the standard deviation varied. For the salt and pepper noise tests, a varying number of pixels were randomly chosen, and set to either 0 or 255. In both cases, viewpoints were treated independently.

The algorithm performs very well when subjected to salt and pepper noise. All error measurements increase linearly with the amount of corrupted pixels, due to the independent nature of the Scene Particles, which prevents a catastrophic failure from occurring, as might be expected with a global approach.

The performance of the system under Gaussian noise is less consistent, although still generally showing a linear increase. In absolute terms, Gaussian noise causes less degradation of the estimated motion than salt and pepper noise, with a standard deviation of 20 intensity values being equivalent to around 15% impulse noise corruption. This is likely due to the multi-scale approach employed, as the smoothing applied to create coarser image scales, also reduces the effect of Gaussian noise.

7 3D Object Tracking

One possible application for accurate and high speed scene flow estimation, is 3D object tracking [2]. Using Expectation Maximization to extract the dominant clusters from the scene particle cloud, provides a data-driven approach to the 3D tracking of objects. This allows automatic detection, segmentation and tracking of moving objects, while avoiding the need for prior knowledge of the object to be tracked.

As a specific example, this approach is applied to tracking hands and heads in 3D during sign language sequences of up to 90 minutes in length. In an unconstrained scenario such as sign language recognition, hands tend to move very rapidly, with sudden changes in trajectory. This makes tracking using the motion field especially suitable, while appearance based tracking may be difficult. Unlike previous approaches [48], skin color assumptions are not necessary in order to provide a segmentation for tracking. Instead an adaptive skin color model was used simply to reduce computational cost by using a smaller number of Scene Particles to estimate the motion of background regions.

The tracking algorithm was applied to a large multiview sign language dataset[2] [49]. No calibration information was provided with the dataset, so the camera parameters were estimated using collection of manually labeled points. This serves to demonstrates that the technique provides some degree of robustness to imprecise calibration. Figure 8 contains examples of the tracked object trajectories, projected onto the input sensors. However, it is important to note that tracking is performed in 3D, and projected to each sensor for display, rather than being performed independently on each video.

Object	Agreement	X RMS error	Y RMS error
Head	100.000%	0.057	0.097
Right Hand	93.535%	0.191	0.100
Left Hand	88.054%	0.277	0.091

TABLE 4: Agreement between projection of estimated 3D trajectories, and 2D trajectories from an alternative system (values in palm widths).

In table 4, the performance of the approach is analyzed compared to the accurate 2D tracker of Pitsikalis *et al.*[50] across 30,000 frames. A frame was labeled as being in agreement, if the estimated positions were less than $\frac{1}{3}$ of a palm width apart. The high level of agreement between the tracking schemes demonstrates the plausibility of the 3D trajectories. Note that the agreement between tracking does not directly provide the accuracy of the system, in fact this could be considered a lower bound on the performance as in some cases, the 2D tracker is in error, due to frontal occlusion, while 3D tracking is maintained using other viewpoints.

In total 2.8 million frames from the dataset were tracked, relating to over 31 hours of 3D sign language trajectories. Using traditional scene flow estimation techniques, an application of this scale would obviously have been intractable (taking roughly 1,600 years to complete). However, it is only one of the possible applications that can now exploit rich 3D motion information, due to the speed of the Scene Particles algorithm.

8 Conclusions

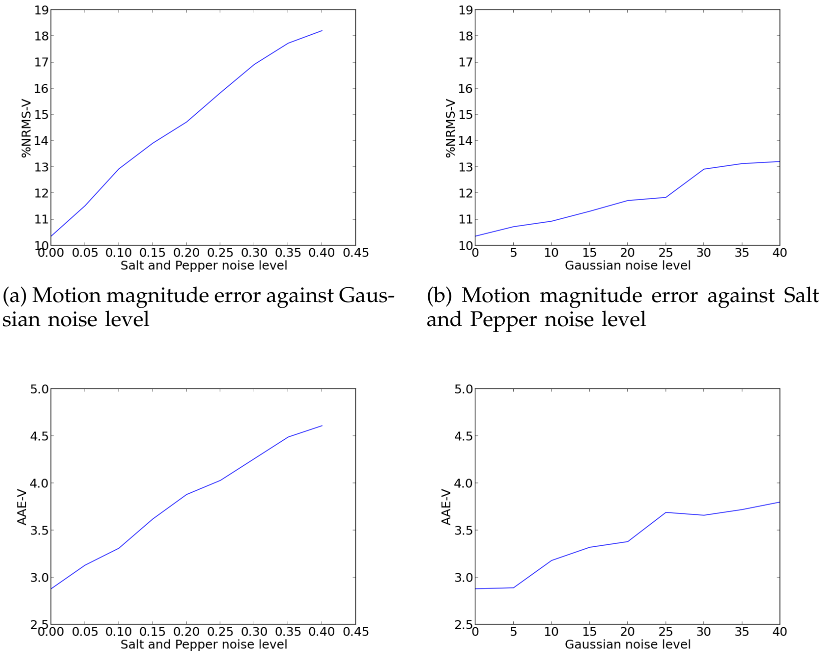
A multi-hypothesis approach to scene flow estimation has been demonstrated, and shown to provide more accurate motion fields than traditional regularized optimization, while also having reduced computational complexity.

Techniques have been demonstrated for applying smoothness constraints to the motion field without compromising the benefits of multiple hypotheses, providing increased accuracy. Additionally, a method for estimating 3D Occlusion maps was presented, further improving scene flow estimation at the cost of increased runtime.

The analysis of the Scene Particle algorithm has highlighted the value of propagating information through time to resolve ambiguities, as opposed to estimating each frame independently. Additionally, the possibility of trading off runtime against accuracy by varying the number of hypotheses, has been demonstrated, and the robustness of the algorithm to noise has been shown.

Finally, an example of the use of scene flow estimation for a traditional vision application has been demonstrated, in which the scene flow field is clustered to obtain the location and velocity of dominant

[2] www.sign-lang.uni-hamburg.de/dicta-sign/portal/



(a) Motion magnitude error against Gaussian noise level

(b) Motion magnitude error against Salt and Pepper noise level

(c) Directional error against Gaussian noise level

(d) Directional error against Salt and Pepper noise level

Fig. 7: Three error measurements, as a function of noise level. The left column shows salt and pepper noise performance, while the right column is the Gaussian additive noise performance. Tests performed using a single appearance and depth sensor, on the sphere dataset.

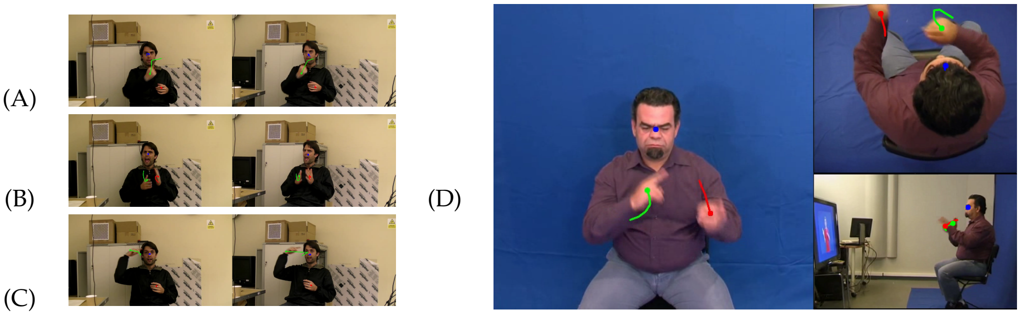


Fig. 8: (A) to (C) are taken from the narrow baseline, 2 view, sequence. (D) Shows a frame from the wide baseline, 3 view, sequence. Discussion is contained in section 7.

objects in the scene. The approach was applied to 3D hand tracking during sign language, and demonstrated to provide excellent performance, despite frequent occlusions and appearance variation. This is only one example of the possible applications, enabled by the speed of the Scene Particles algorithm. Future work is to provide a CUDA implementation which we expect to allow accurate scene flow estimation to be performed in real-time.

Acknowledgments

The work presented has been supported by the European Community's Seventh Framework Programme (FP7/2007-2013) grant agreement no 231135 (Dicta-Sign), and by the EPSRC project "Learning to Recognise Dynamic Visual Content from Broadcast Footage" (EP/I011811/1).

References

[1] H. Yan and T. Tjahjadi, "Optical flow estimation and segmentation through surface fitting and robust statistics," in *SMC*, 2003.

[2] S. Hadfield and R. Bowden, "Go with the flow: Hand trajectories in 3d via clustered scene flow," in *ICIAR*, 2012.

[3] R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition," in *FG*, 1998.

[4] B. Fransen, E. Herbst, A. Harrison, W. Adams, and J. Trafton, "Real-time face and object tracking," in *Intelligent Robots and Systems*, 2009.

[5] R. Cipolla, Y. Okamoto, and Y. Kuno, "Robust structure from motion using motion parallax," in *ICCV*, 1993.

[6] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *ICCV*, 2003.

[7] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *Int. Symp. Wearable Computers*, 2008, pp. 15–22.

[8] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *ICCV*, 1999.

[9] S. Vedula, S. Baker, S. Seitz, and T. Kanade, "Shape and motion carving in 6d," in *CVPR*, 2000.

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

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

[12] T. Schuchert, T. Aach, and H. Scharr, "Range flow for varying illumination," in *ECCV*, 2008.

[13] ——, "Range flow in varying illumination: Algorithms and comparisons," *PAMI*, 2009.

[14] T. Lukins and R. Fisher, "Colour constrained 4d flow," in *Proc BMVC*, 2005.

[15] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, "Efficient dense scene flow from sparse or dense stereo data," in *Proc. ECCV*, 2008.

[16] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *IJCV*, 2011.

[17] J. Pons, R. Keriven, O. Faugeras, and G. Hermosilo, "Variational stereovision and 3d sceneflow estimation with statistic similarity measure," in *ICCV*, 2003.

[18] J. Pons, R. Keriven, and O. Faugeras, "Multiview stereo reconstruction and scene flow estimation with a global image-based matching score," *IJCV*, 2007.

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

[20] S. Hadfield and R. Bowden, "Kinecting the dots: Particle based scene flow from depth sensors," in *Proc ICCV*, 2011.

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

[22] Y. Furukawa and J. Ponce, "Dense 3d motion capture from synchronized video streams," in *CVPR*, 2008.

[23] J. Courchay, J. Pons, P. Monasse, and R. Keriven, "Dense and accurate spatio-temporal multi-view stereovision," in *ACCV*, 2009.

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

[25] 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 *Proc. ICCV*, 2001.

[26] Y. Zhang and C. Kambhamettu, "Integrated 3d scene flow and structure recovery from multiview image sequences," in *CVPR*, 2000.

[27] M. Isard and J. MacCormick, "Dense motion and disparity estimation via loopy belief propagation," in *ACCV*, 2006.

[28] R. Li and S. Sclaroff, "Multi-scale 3d scene flow from binocular stereo sequences," in *Workshop Motion and Video Computing*, 2005.

[29] Y. Zhang and C. Kambhamettu, "On 3d scene flow and structure estimation," in *CVPR*, 2001.

[30] ——, "On 3-d scene flow and structure recovery from multiview image sequences," *SMC*, 2003.

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

[32] J. Ruttle, M. Manzke, and R. Dahyot, "Estimating 3d scene flow from multiple 2d optical flows," in *International Machine Vision and Image Processing Conference*, 2009.

[33] T. Basha, S. Avidan, A. Hornung, and W. Matusik, "Structure and motion from scene registration," in *CVPR*, 2012.

[34] D. Sun, S. Roth, J. Lewis, and M. Black, "Learning optical flow," in *ECCV*, 2008.

[35] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing*, 1993.

[36] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *IJCV*, 1998.

[37] P. Del Moral, *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer, 2004.

[38] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis*, 2005.

[39] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *ICCV*, 2007.

[40] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *IJCV*, 2012.

[41] H. Sidenbladh, "Probabilistic tracking and reconstruction of 3d human motion in monocular video sequence," Ph.D. dissertation, Stockholm Royal Institute of Technology, 2001.

[42] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt, "Joint estimation of motion, structure and geometry," in *ECCV*, 2010.

[43] M. Gong and Y.-H. Yang, "Disparity flow estimation using orthogonal reliability-based dynamic programming," in *IAPR International Conference on Pattern Recognition*, 2006.

[44] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *CVPR*, 2003.

[45] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *IVCNZ*, 2008.

[46] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *PAMI*, 2011.

[47] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *PAMI*, 2008.

[48] H.-P. Huang and C.-T. Lin, "Multi-camshift for multi-view faces tracking and recognition," in *ROBIO*, 2006.

[49] E. Efthimiou, S.-E. Fotinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and F. Lefebvre-Albaret, "Sign language technologies and resources of the dicta-sign project," in *Workshop on the Representation and Processing of Sign Languages*, 2012.

[50] V. Pitsikalis, S. Theodorakis, C. Vogler, R. Athena, and P. Maragos, "Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition," in *Workshop on Gesture Recognition*, 2011.

Simon Hadfield received his PhD from the University of Surrey in 2013 where he also received an MEng (distinction) in Electronic and Computer Engineering in 2009. He has been awarded the DTI MEng Prize, for the top performance by an MEng graduate, and the Associateship of the University of Surrey (AUS) for his industrial placement with the Home Office Scientific Development Branch. He is a research fellow at the Centre for Vision Speech and Signal Processing at the University of Surrey. His research interests include motion estimation, gesture recognition and HCI.

Richard Bowden received a BSc degree in computer science from the University of London in 1993, an MSc (distinction) from the University of Leeds in 1995, and a PhD degree in computer vision from Brunel University in 1999 for which he was awarded the Sullivan Doctoral Thesis Prize for the Best UK PhD thesis in vision. He is currently a Professor of computer vision and machine learning at the University of Surrey, United Kingdom, where he leads the Cognitive Vision Group within the Centre for Vision Speech and Signal Processing and was recently awarded a Royal Society Leverhulme Trust Senior Research Fellowship. His research centers on the use of computer vision to locate, track, and understand humans. His research into tracking and artificial life received worldwide media coverage, appearing at the British Science Museum and the Minnesota Science Museum. He was a member of the British Machine Vision Association (BMVA) executive committee and a company director for seven years. He is a member of the BMVA, a fellow of the Higher Education Academy, and a senior member of the IEEE.