Structure and Motion from Scene Registration

Tali Basha Shai Avidan
Tel Aviv University
{talib, avidan}@eng.tau.ac.il

Alexander Hornung
Disney Research Zurich
hornung@disneyresearch.com

Wojciech Matusik
MIT CSAIL
wojciech@csail.mit.edu

Abstract

We propose a method for estimating the 3D structure and the dense 3D motion (scene flow) of a dynamic nonrigid 3D scene, using a camera array. The core idea is to use a dense multi-camera array to construct a novel, dense 3D volumetric representation of the 3D space where each voxel holds an estimated intensity value and a confidence measure of this value. The problem of 3D structure and 3D motion estimation of a scene is thus reduced to a nonrigid registration of two volumes–hence the term "Scene Registration". Registering two dense 3D scalar volumes does not require recovering the 3D structure of the scene as a pre-processing step, nor does it require explicit reasoning about occlusions. From this nonrigid registration we accurately extract the 3D scene flow and the 3D structure of the scene, and successfully recover the sharp discontinuities in both time and space. We demonstrate the advantages of our method on a number of challenging synthetic and real data sets.

1. Introduction

Structure and motion estimation from image data is of fundamental importance in many vision and graphics applications, including 2D optical flow for image-based rendering, nonrigid volumetric registration of medical data sets, object tracking, navigation, or virtual reality.

Our objective is to recover the 3D structure and the motion of a nonrigid 3D scene, captured with a calibrated dense camera array (i.e., the baseline between each pair of adjacent cameras is small, see Fig.1). This problem received considerable attention in the last decade and various algorithms have been proposed to solve it. These algorithms use multiple cameras to estimate the *scene flow*, where scene flow is defined as the dense 3D motion field of a nonrigid scene [24]. It follows directly from this definition that 3D recovery of the surface must be an essential part of scene flow algorithms, unless it is given a priori.

Most existing methods for 3D structure and scene flow estimation require establishing dense correspondence between pixels or regions of the same scene taken from different views at different time steps. The correspondence problem brings with it several classical challenges, including ambiguities due to a small field of view or low texture regions (the aperture problem), dissimilar appearance of the scene over time or from different views, and image noise. Another central difficulty in estimating dense correspondence fields is the occlusion problem. That is, regions visible in one image but having no counterparts in other, and hence cannot be matched.

a **b**

Figure 1. **(a)**, The ProFusion 5x5 camera array. The central camera is the reference. **(b)**, Linear stage with a Canon 5D Mark II DSLR.

Extensive research has been carried out to address the problem of correspondence in time and space, mostly in the area of optical flow and stereo estimation. Most existing methods for scene flow or multi-view reconstruction rely on a "photo-consistency" measure for evaluating the visual compatibility of the correspondence across multiple images. In addition, most of these methods define a "visibility model", to determine the occlusion relationship between the recovered 3D points. The visibility model determines which images should participate in the estimation of the photo-consistency measure. The photo-consistency measure and the visibility model interfere with each other because errors in the photo-consistency measure affect the visibility modeling and vice-versa. Thus, despite many advances in recent years [3, 18] handling occlusions and the resulting discontinuities is still an open research problem.

Our approach sidesteps the need to recover or handle occlusions and does not require explicit reasoning about flow discontinuities, a requirement that adversely affects scene

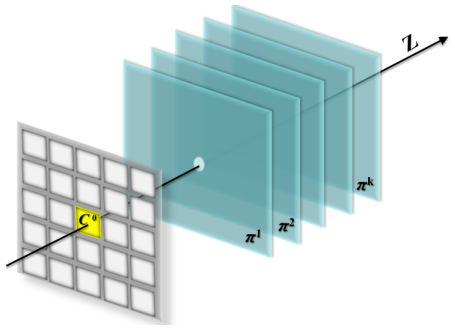


Figure 2. The 3D space is discretized using a set of K fronto-parallel planes, $\{\pi^k\}_{k=1}^K$, with respect to the central camera, C^0.

flow methods. We represent the problem in a 3D volumetric space that does not distinguish between real and free 3D scene points. Specifically, we use a dense, calibrated and synchronized multi-camera array to capture a dynamic non-rigid 3D scene at two time steps. The set of images captured by the camera array at each time step samples the light rays in the scene. The captured data at each time step is then represented with a discretized 3D volume, where each cell stores the distribution of light rays passing through it. We approximate this 3D vector field of light rays by a 3D scalar volume. In particular, the set of light rays at each point is reduced to a scalar and a confidence measure, estimated by nonlinear filtering.

The proposed approach has a number of benefits. First, we sidestep the need to reason about visibility, which is taken care of automatically in the volumetric registration step. Second, each voxel in our volume is assigned with our confidence that there is a real 3D point there. This way we do not have to commit to the 3D structure. Thus, we can compute the flow between the two volumes, which represent the scene at two time steps, without explicit representation of the scene. Computing the flow in this volumetric space amounts to matching two 3D scalar fields, a problem that has been addressed in the past. Finally, the method is scalable and, to the best of our knowledge, we are the first to compute both 3D structure and scene flow from tens of cameras.

Once correspondence in this space is estimated, we use the confidence measure to extract both the scene flow and the 3D structure, and successfully estimate the sharp discontinuities in both time and space.

2. Background

Due to the considerable body of work on flow and reconstruction, we focus on research we consider most relevant to ours. See [3, 18] for a recent overview and evaluation on optical flow and multi-view reconstruction.

The seminal work of Horn and Schunck [9] on optical flow estimation assumed a smooth deformation field. This works well in many cases but fails around boundaries of independently moving image regions. Since then a large body of work focused on the problem of flow discontinuities [17, 5, 7, 2, 1, 26, 20].

Using more than a single camera enables the estimation of a three dimensional scene flow, as opposed to the two dimensional optic flow which is simply the projection of the scene flow onto the image plane of a camera. Vedula *et al*. [24] compute the 3D scene flow from multiple images by lifting the 2D optical flow between several image pairs to 3D. However, they do not enforce consistency across the flow fields. This was later addressed by Huguet *et al*. [10] and Basha *et al*. [4] that simultaneously recover the depth and motion. Some of the work on scene flow estimation assumed a 2D parametrization of the problem [25], but recently there is a growing body of literature that uses a 3D parametrization for solving both 3D structure and 3D motion [14, 16].

Our work is also closely related to the problem of voxel coloring, where the goal is to reconstruct a *static* 3D structure from multiple cameras [19]. Voxel coloring discretizes the space into voxels and determines for each voxel if it is occupied or not based on the mutual agreement of its projection on the images, as well as occlusion reasoning. Voxel coloring assumes a certain arrangement of cameras and this assumption was later removed by Space Carving [13] that still makes hard decisions about voxel occupancy. Space Carving was extended to deal with probabilistic space carving [6] as well as non Lambertian surfaces [27, 22] but all extensions depends on photo-consistency and need to deal occlusions. One way to adopt voxel coloring, or any of its descendants, for our needs is to estimate the 3D structure at each time step and then estimate the scene flow between the two. Unfortunately, there is no guarantee that the 3D structure estimations will be consistent and hence the scene flow estimation might fail, as we show in the experimental section.

Neumann *et al*. [15] proposed a new camera design to deal with 3D motion estimation (as opposed to 3D reconstruction). They first define a polydioptric camera which is a generalized camera that captures a multi-perspective subset of the space of light rays (dioptric: assisting vision by refracting and focusing light). And then show that 3D motion estimation of polydioptric cameras based on a light field representation is independent of the actual scene. The intensity of a light ray does not change over time in a static scene with fixed illumination and therefore matching light rays is possible.

We, on the other hand, consider a dynamic scene where the correspondence between light rays is ill defined since the intensity of a light ray can change due to nonrigid deformations. So instead of matching individual light rays, we match 3D points where each point aggregates the distribu-

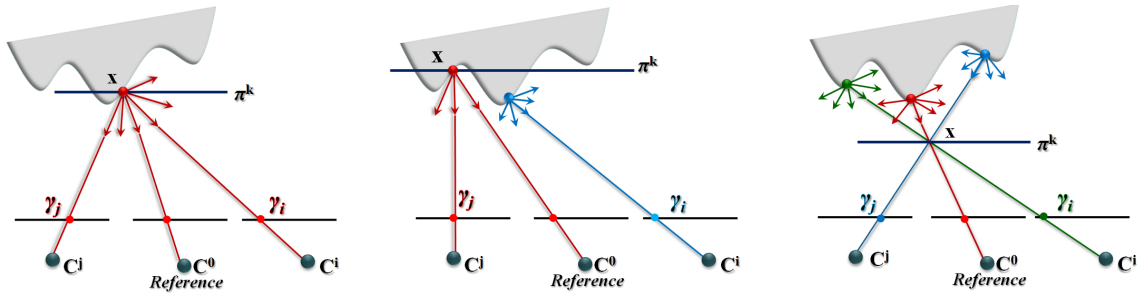


Figure 3. 2D illustration of the volume representation; each point in the volume is represented by the set of intensities that are captured by the camera array; shown are the intensities for of real scene point **(a)** that is visible in all cameras, for a point that is partially occluded **(b)**, and for point in the free space **(c)**.

tion of light in multiple directions. This makes our approach more robust to handling dynamic scenes.

3. The Method

We use a multi-camera array, consisting of N cameras, to capture a dynamic non rigid scene at two different time steps. The cameras are assumed to be calibrated, synchronized, and each pair of adjacent cameras is assumed to have small baseline.

Each set of images, captured at a single time step, is used to construct a 3D volume, $V(\mathbf{x})$, where every cell holds a 2D distribution of the light rays that pass through the point $\mathbf{x}$. We then approximate $V(\mathbf{x})$ to obtain a scalar volume, $S(\mathbf{x})$, by applying a nonlinear filter to the captured light rays at each scene point. In this scalar volume, occluding surfaces are blurred out in the vicinity of the objects boundaries. Furthermore, $S(\mathbf{x})$, which consist of real scene points as well as points in the free space, is a piecewise continuous representation with respect to all three dimensions (x, y and z). This lets us perform a dense matching of the two scalar volumes, S^t and S^{t+1}, computed at two time steps, prior to recovering the 3D structure of the scene. By doing so, we bypass the need to reason about occlusions or sharp discontinuities in both the 3D structure and 3D motion field.

Finally, the computed flow between S^t and S^{t+1} is used to extract both the 3D structure and the 3D motion and to recover the sharp discontinuities in both the depth and the motion field. In the following we first describe the construction of S. Section 3.2 then describes how the volumes S^t and S^{t+1} computed at two consecutive time steps can be registered.

3.1. 3D Representation

We discretize the 3D space using a set of K fronto-parallel planes, $\{\pi^k\}_{k=1}^K$, with respect to the central camera, C^0 (see Fig. 2). For each plane, the images from all N cameras are aligned to the reference view by computing the homographies between the views, in a way similar to the stereo parametrization used by Szeliski & Golland [21].

Formally, let $V(\mathbf{x})$ be the volume:

$$V(\mathbf{x}) = \{\gamma_i \mid 0 \leq i \leq N-1\}, \tag{1}$$

where γ_i is the intensity of the pixel that is the projection of the 3D point $\mathbf{x}$ onto the i^{th} camera (see Fig. 3 for 2D illustration). This gives us a 2D distribution function of the light rays passing through the point $\mathbf{x}$ and sampled by the camera array. In practice, the 3D space is sampled by back-projecting each pixel in the reference view onto each of the planes, $\{\pi^k\}_{k=1}^K$. Namely, there is a known transformation between each volume point, $\mathbf{x}$, to a pixel in the reference view.

Observe that γ_i represents the true intensity of the 3D point only if it is a real scene point that is visible to the i^{th} camera (see Fig. 3(a)). In case the point is occluded in the i^{th} camera (see Fig. 3(b)) or in case it is a point in free space, (see Fig. 3(c)), then γ_i is the intensity of an arbitrary point.

Then why is $V(\mathbf{x})$ a useful representation? To understand this, consider the simple case of a scene that consists of Lambertian background and foreground objects captured by a dense camera array. A scene point that is visible in all views would have a unimodal distribution centered around the surface irradiance at that point. A 3D point that is located in free space will not have such a unimodal distribution because random 3D points are projected to the various cameras in the camera array. A 3D point that is occluded in all views will behave in the same way a free 3D point would (see Fig. 4). This reasoning should hold for scenes with more objects as well. We will take advantage of the fact that only real scene points are supposed to have a unimodal distribution.

In order to allow for an efficient and robust correspondence estimation, we reduce the 2D distribution of rays at each 3D point in $V(\mathbf{x})$ to a single scalar $S(\mathbf{x})$, which allows us to use existing 3D matching techniques for computing the 3D motion of every point in the 3D space.

Simple averaging of samples from all cameras, as is done for example in voxel coloring [19] or synthetic aperture

photography [11], results in combining intensities of different 3D points in the scene. In particular, scene points that are occluded in some of the cameras are likely to be assigned a different intensity; hence, the matching between the volumes is prone to errors. Instead, we take a more robust measure that assigns a coherent intensity to scene points, while blurring out points in the free space. We demonstrate the advantages of our method over simple averaging in the experimental section.

Following the discussion above, we assume that the largest mode (i.e., the most frequent intensity value) of $V(\mathbf{x})$ corresponds to the true irradiance of $\mathbf{x}$, in the case that $\mathbf{x}$ is a scene point. However, if $\mathbf{x}$ is a point in free space, or is completely occluded, choosing the intensity to be one of the modes results in random noise. Fig. 4 shows a typical distribution of light rays, in terms of their corresponding gray level histogram, for three pixels that correspond to a visible scene point, partially occluded scene point and a point in free space.

Given $V(\mathbf{x})$ we wish to compute volume $S(\mathbf{x})$ and confidence $C(\mathbf{x})$. Formally, we detect the modes of $V(\mathbf{x})$ for every point $\mathbf{x}$ using QuickShift [23] and obtain the following:

$$\mathbf{m}(\mathbf{x}) = (m_1, m_2, ...)^T, \tag{2}$$

$$\mathbf{n}(\mathbf{x}) = (n_1, n_2, ...)^T, \tag{3}$$

where $\mathbf{m}(\mathbf{x})$ is the vector (of variable size) of the intensity centers of the detected modes; $\mathbf{n}(\mathbf{x})$ is the vector of the cardinality of each mode (i.e., how many samples belong to each mode). We denote m_* and n_* to be the intensity and cardinality of the largest mode.

We determine the intensity of each 3D point, $\mathbf{x}$, by averaging the modes of $V(\mathbf{x})$:

$$S(\mathbf{x}) = \widehat{\mathbf{c}}(\mathbf{x})^T \mathbf{m}(\mathbf{x}), \tag{4}$$

where $\widehat{\mathbf{c}}(\mathbf{x}) = (\widehat{c}_1, \widehat{c}_2, ...)$ is a vector of weights. We set the weight $\widehat{c}_i$ of mode i, according to its cardinality, n_i, and deviation from the intensity of the largest mode, m_*:

$$\widehat{c}_i = \frac{n_i}{N_{\mathbf{x}}} \left(\mu \frac{1}{\sqrt{(m_i - m_*)^2 + \epsilon}} + 1 - \mu \right), \tag{5}$$

where $N_{\mathbf{x}} = \sum_i n_i$ is the total number of cameras that view the point $\mathbf{x}$ and $\epsilon = 0.001$. To control the relative impact of the intensity deviation, we set $\mu = \frac{n_*}{N_{\mathbf{x}}} \in [0, 1]$. It follows that a large mode which is close to m_*, will have high weight. We found this heuristic choice to work well in practice.

A confidence measure is computed for each point, $\mathbf{x}$, by taking the ratio of the number of cameras in the largest mode of the distribution compared to the total number of cameras that view that point:

$$C(\mathbf{x}) = \mu = \frac{n_*}{N_{\mathbf{x}}}. \tag{6}$$

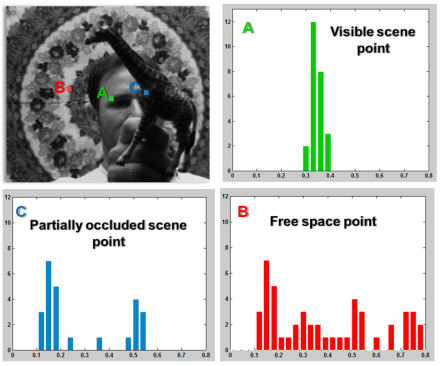


Figure 4. The histograms of the gray level values, as captured from a twenty-five camera array, for: (A), a visible scene point, (B), point in the free space, and (C), partially occluded scene point.

This way scene points will have a high confidence, because we expect their distribution to be unimodal, while all other points (i.e. those in free space or those that are occluded) will have a very low confidence.

Following this procedure, each sampled volume $V(\mathbf{x})$ is reduced to a scalar-valued 3D volume $S(\mathbf{x})$, and the corresponding confidence $C(\mathbf{x})$. Now we are able to compute the matching between the volumes computed at two successive time steps t and $t+1$.

3.2. 3D Registration

Each of the two 3D scalar volumes S^t and S^{t+1}, can be regarded as a sampling of a piecewise continuous volume with respect to all three dimensions (x,y, and z). This property enables us to find the matching between the volumes using nonrigid 3D registration techniques. In particular, we use the method of Glocker *et al.* [8], previously used in the context of medical imaging. This method allows the brightness constancy assumption to be imposed between the source and target volumes (data term) and the smooth-

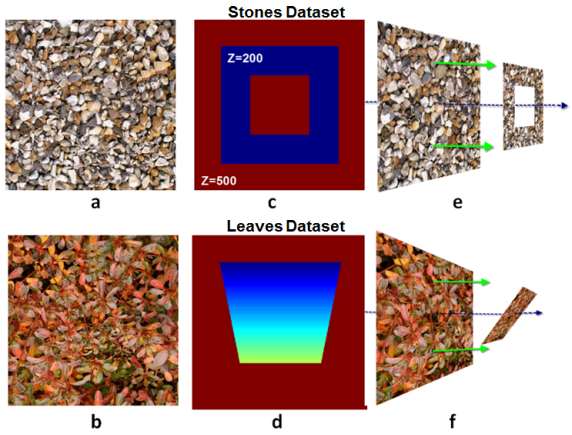
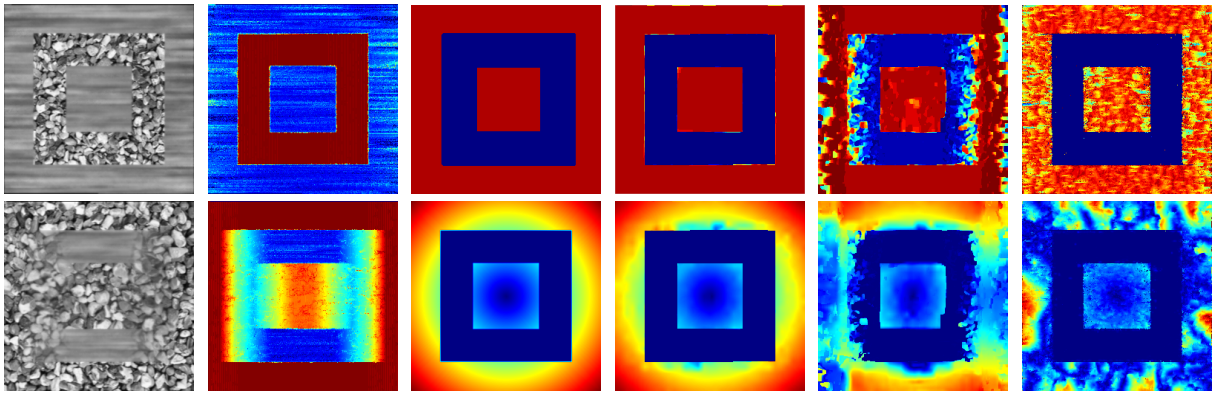


Figure 5. **Synthetic datasets: (a-b)**, the reference view of each dataset; **(c-d)**, the corresponding depth maps; **(e-f)**, side view of the scene. The background translates in the depth direction.



a (Image) **b** (Confidence) **c** (Ground Truth) **d** (Robust) **e** (Average) **f** (Voxel Coloring)

Figure 6. **Ground truth evaluation:** The top row **(b-f)** shows depth results of a volume slice (frontoparllel plane) focused on the foreground. The bottom row **(b-f)** shows optical flow results of a volume slice focused on the background. **(a)** Slices of the volume focused on the foreground (top) and background (bottom). **(b)** The computed confidence map; high confidence points are colored in red; **(c)** Ground truth of depth (top) and optical flow magnitude (bottom); **(d)** Robust scene registration; **(e)** Simple averaging (instead of robust) scene registration; **(f)** Robust voxel coloring.

ness assumption on the 3D flow (smoothness term), using arbitrary cost functions. A global objective functional is defined to express these assumptions and then is reformulated as a discrete labeling problem. In order to account for large displacements and to achieve sub-pixel accuracy, a multi-scale incremental approach is considered where the optimal solution is iteratively updated. The discretized functional at each level is efficiently minimized using the Markov Random Field (MRF) optimization method of [12].

To use the method of Glocker *et al.*[8] let $\mathcal{F}(\mathbf{x})$ denote the 3D flow between $S^t(\mathbf{x})$ and $S^{t+1}(\widehat{\mathbf{x}})$. That is,

$$\mathcal{F}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x})), \tag{7}$$

where u, v and w are the flow's horizontal, vertical and depth components, respectively. We chose the data cost to be the weighted sum between the brightness constancy and gradient constancy assumptions. That is,

$$E_{Data}(\mathcal{F}) = \frac{1}{|\Omega_S|}\left((1-\lambda)\sum_{\mathbf{x}\in\Omega_S}|S^t(\mathbf{x}) - S^{t+1}(\widehat{\mathbf{x}})| - \lambda\sum_{\mathbf{x}\in\Omega_S}\left|\frac{\nabla S^t(\mathbf{x})}{|\nabla S^t(\mathbf{x})|}\cdot\frac{\nabla S^{t+1}(\widehat{\mathbf{x}})}{|\nabla S^{t+1}(\widehat{\mathbf{x}})|}\right|\right) \tag{8}$$

where $\widehat{\mathbf{x}} = \mathbf{x} + \mathcal{F}(\mathbf{x})$, λ controls the relative weight between the terms and Ω_S denotes the volume domain.

The smoothness term is given by the truncated L_1 distance between the flow of neighboring volume points:

$$E_{Smooth}(\mathcal{F}) = \sum_{\mathbf{x}\in\Omega_S}\sum_{\mathbf{x}_n\in\mathcal{N}(\mathbf{x})} min\{||\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_n)||_1, \eta\}, \tag{9}$$

where $\mathcal{N}(\mathbf{x})$ is the neighborhood of point $\mathbf{x}$, and η is the truncation threshold. In practice, a truncation of the smoothness term allows discontinuities in the flow $\mathcal{F}(\mathbf{x})$.

3.3. Depth and Optical Flow Estimation

After the volumetric registration stage, every point in the volume is associated with an intensity value, a confidence measure, and a 3D flow estimation. With these in hand, we extract both the 3D structure and 3D motion of the scene (scene flow) w.r.t the reference camera by assigning a depth value for each pixel in the reference camera. That is, once the depth value of each pixel is chosen, the optical flow and depth value at the following time step are directly determined by the computed 3D flow.

Each pixel $p = (x, y)$ in the reference camera I^0 is associated with a set of K possible locations in the volume, given by: $\{\mathbf{x^i} = (x, y, z^i)\}_{i=1}^K$, where $\{z^i\}_{i=1}^K$ are the discretized depth values that form the volumetric space. The intensity, the confidence and the 3D flow of $\mathbf{x^i}$ are respectively given by:

$$\{S^t(\mathbf{x^i}), C^t(\mathbf{x^i}), \mathcal{F}(\mathbf{x^i})\}. \tag{10}$$

We wish to find the optimal depth z^* that minimizes the difference between the intensity, $I^0(p)$, of pixel p in reference image I^0 and the intensity $S^t(x, y, z^*)$ in the volume. We also assume that the optimal assignment should have high confidence, so we want $C^t(x, y, z^*)$ to be high. These assumptions are formulated as a MRF multi-labeling optimization problem where a label assignment ℓ_p associates the pixel p with the point, $\mathbf{x}^{\ell_\mathbf{p}}$. Formally, the data term is defined as a weighted sum of the above mentioned assumptions and is given by:

$$E_{Data}(\mathbf{L}) = \sum_{p\in\Omega}|I^0(p) - S^t(\mathbf{x}^{\ell_p})| + \alpha(1 - C^t(\mathbf{x}^{\ell_p})), \tag{11}$$

where $\mathbf{L}$ is the set of discrete assignments of all pixels, α controls the relative impact of each of the terms, and Ω denotes the reference camera domain.

A spatial smoothness term is added, expressing the assumption that neighboring pixels have similar depth values. That is,

$$E_{Smooth}(\ell) = \sum_{p\in\Omega}\sum_{q\in\mathcal{N}(p)} |\ell_p - \ell_q|, \quad (12)$$

where $\mathcal{N}(p)$ is the neighborhood of pixel p. The total energy,

$$E(\ell) = E_{Data}(\ell) + \beta \cdot E_{Smooth}(\ell), \quad (13)$$

is effectively minimized using graph cuts. We use a fairly low value of β in our implementation.

Finally, given that the optimal assignment for each p is $z(\ell_p^*)$, we define $\mathbf{x}^* = (x, y, z(\ell_p^*))$. The optical flow of p is given by $(u(\mathbf{x}^*), v(\mathbf{x}^*))$, and the new depth value at the following time step is given by $(z^* + w(\mathbf{x}^*))$. The computed optical flow and the depth maps at two time steps can now be reprojected (using the camera intrinsic parameters) in order to recover the exact 3D structure and 3D motion in the perspective 3D space.

Observe that we were able to extract the optical flow and the 3D structure without reasoning about visibility of 3D points in the cameras. This is the key insight in extracting the depth and optical flow from our volumetric space.

4. Experiments

We conducted a number of experiments on synthetic and real data to evaluate several aspects of the proposed method. First, we compare ourselves to other, state-of-the-art methods on synthetic data and find that we are more accurate. Second, we show that our method can handle sharp discontinuities in both shape and motion on both synthetic and real-world scenes. In addition, in one experiment we use 1D camera array of 100 cameras to demonstrate the scalability of our method.

We also analyze two key design decisions that we made. The first decision we analyze is the reduction of the vector-valued volume $V(\mathbf{x})$ to scalar volume $S(\mathbf{x})$ using QuickShift, as opposed to simple averaging. We find that QuickShift is more robust and leads to better overall accuracy (see Fig.6**(e)**). The second design decision we analyze is the use of our scene registration, i.e., performing a matching prior to recovering the 3D structure of the scene. To this end, we use a robust version of Voxel coloring, where photo consistency is estimated using QuickShift, to estimate 3D structure at each time step independently and then register the two volumes. We find that the results are not as good as our method, because Voxel coloring does not guarantee consistent 3D structure at both time steps, an inconsistency that adversely affects the registration step.

Ground Truth Evaluation: We tested our method on two challenging synthetic scenes that were rendered in OpenGL. The scenes are viewed by a dense 1D array of 51 cameras, and consist of a moving foreground that is placed at a distance of *Z=200* in front of a background plane that is located at *Z=500*, (the units are arbitrary). The foreground, i.e., a frontoparallel frame in the first scene and a tilted plane in the second scene, is moving 70 units in the depth direction w.r.t. the reference camera (See Fig. 5). Therefore, large discontinuities and occlusions are introduced in both the spatial and the temporal domains. In both experiments the depth was discretized into twenty-five levels, and the reference camera was the central one.

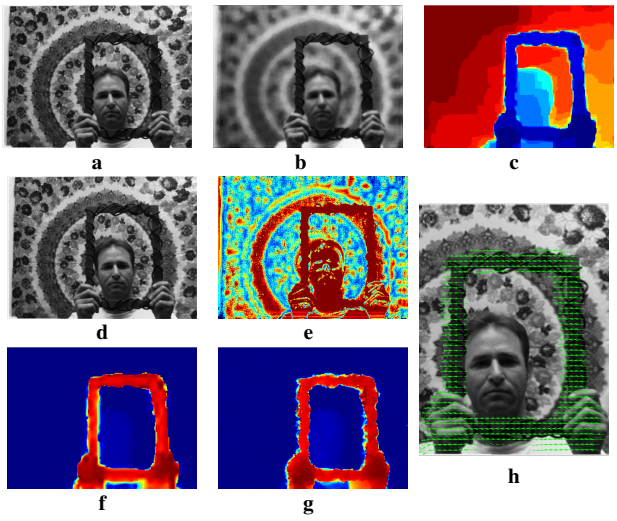


Figure 7. **Real dataset (Frame): (a,d)** the reference view at time t and $t+1$, respectively; **(b)** the slice from our 3D volume that corresponds to foreground object; **(e)** the computed confidence map of **(b)**, with high confidence points colored in red; **(c)** the estimated depth map. **(f)** the magnitude of the optical flow estimated by our method; **(g)** the magnitude of the optical flow estimated by [28]. **(h)** the flow map of our optical flow.

The results for the first experiment are presented in Fig. 6. As Fig.6.**(d)** clearly demonstrates, we successfully obtain accurate results for both the optical flow and the

		RMS		*AAE*
		u	v	*(deg)*
Stones	Our method (N=51)	0.57	0.69	2.8
	Our method (N=25)	0.60	0.78	3.25
	Our method (N=7)	0.68	0.79	3.34
	S.F. [4]	1.32	1.80	3.32
	O.F. [28]	2.03	1.86	5.83
Leaves	Our method (N=51)	0.38	0.47	1.70
	Our method (N=25)	0.39	0.48	1.70
	Our method (N=7)	0.57	0.53	1.98
	S.F. [4]	1.14	1.26	1.83
	O.F. [28]	1.31	1.41	3.67

Table 1. The evaluated errors (w.r.t ground truth) of the extracted optical flow computed with our method and comparison to the projection of the scene flow results of Basha *et al.*[4] and the optical flow results of Zach *et al.*[28].RMS error in the optical flow, (u, v). Also shown is the absolute angular error (AAE).

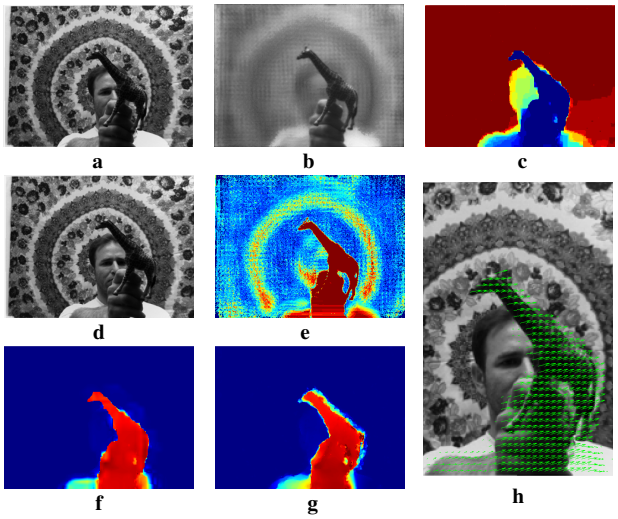


Figure 8. **Real dataset (Giraffe):** Please see caption of Figure 7 for the description of the subfigures.

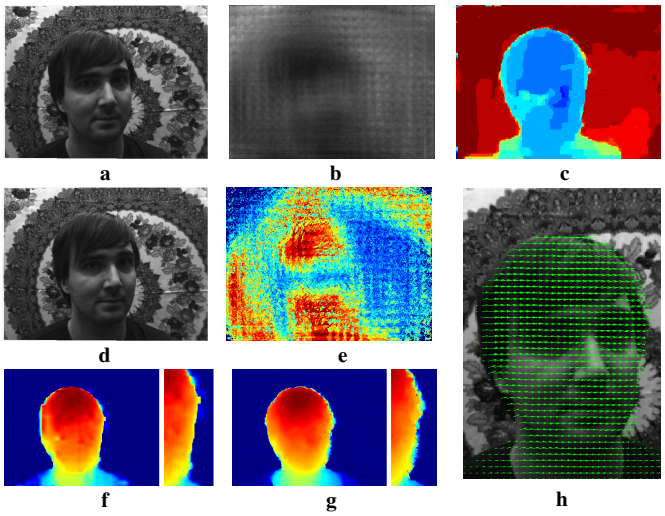


Figure 9. **Real dataset (Simon):** Please see caption of Figure 7 for the description of the subfigures. The narrow figures to **(f-g)** is a close-up of the boundary region.

depth. We quantitatively evaluated our results by computing the *RMS* and the *AAE* with respect to the ground truth. To test the affect of the number of cameras on accuracy, we evaluated our results with a smaller number of cameras (N=25 and N=7). The computed errors of our results compared to the errors of the multi-view scene flow method of Basha *et al.*[4][1], and to the optical flow method of Zach *et al.*[28] are summarized in Table 1, which demonstrates the higher accuracy of our method.

Real Data: We tested our method on several real-world sequences that were captured by the 5×5 camera array, ProFUSION 25 (see Fig. 1(a)). The cameras are arranged with 12mm spacing and provide 640×480 images of raw data at a rate of 25FPS. Due to the narrow baseline setup, the camera array was placed at a distance of 1.5-2 meters from the background. The cameras were calibrated independently using OpenCV and the images were corrected for lens distortion and vignetting. In all the real-data experiment the depth was discretized into thirty values and the images were downsampled by a factor of two.

Our results for three datasets are presented in Fig. 7-9. The first two datasets demonstrate large discontinuities in depth and motion. In the second dataset (Fig. 8), larger motion is considered and hence significant occlusions in temporal domain must be dealt with.

Fig. 7-9, show the recovered depth and the magnitude of the estimated optical flow, for each of the first two datasets. In addition, we present the 2D slice from the volume that corresponds to the foreground object and its associated confidence map. As can be seen, the foreground object is in focus while the rest of the scene is blurred out. Moreover, the foreground object is assigned a high confidence as expected.

[1]Due to the high computational complexity of [4] the results were computed from seven input views.

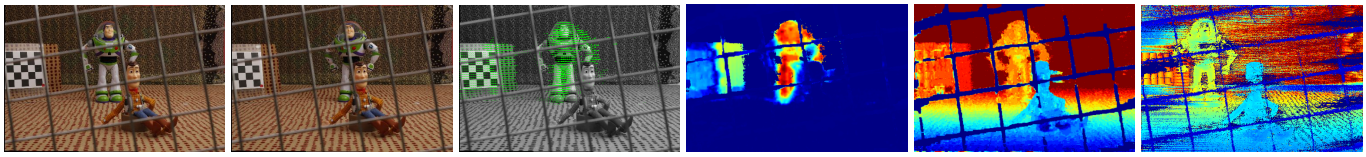
However, a closer look shows that there are additional high confidence regions that belong to the background. The reason is the low variance of intensities in those regions. In particular, these regions cannot be distinguished during the clustering stage (despite the wrong depth value), and hence, the 3D structure cannot be obtained from the confidence alone. Nonetheless, since the depth and optical flow are extracted using the brightness constancy assumption (between the volume and the reference camera) as well, we successfully obtain the correct solution.

The third dataset (Fig. 9) involves nonrigid motion of a moving face. The recovered depth map shows that the depth differences between parts of the face are recognized. This is also shown in Fig. 9.(f), where the nose is assigned low confidence.

In the last experiment we used a camera stage (see Fig.1(b)) to capture a scene with several toys that move behind a wire fence. At each time step we take 100 images, of size 320×240, of the scene. Fig. 5 shows the results of this experiment. As can be seen, we successfully recovered the 3D motion of the toys, as well as the 3D structure of the scene, despite sharp discontinuities (for example, the wire fence).

5. Conclusions

Scene registration is a method for computing the structure and motion of a dynamic nonrigid scene, captured with a camera array. A feature of our approach is that it does not require explicit occlusion handling and improves the reconstruction of discontinuities both in space and time. The key idea of our method is to convert the input sets of images into a novel volumetric space. In this volume both real scene points and points in free space are represented by a



a **b** **c** **d** **e** **f**

Figure 10. **Real dataset (Toy Story)**: **(a),(b)** the reference view at time t and $t+1$, respectively; Buzz, as well as the check-board, moved; **(c)** the flow map of our optical flow; **(d)** the magnitude of our optical flow; **(e)** our depth map; **(f)** the estimated depth map using robust Voxel Coloring.

scalar value and a confidence measure. With this representation the flow computation is reduced to a nonrigid registration of two 3D scalar volumes that does not require explicit reconstruction of the 3D scene structure or reasoning about occlusions. Instead, the scene flow and structure can be recovered easily after the volumetric registration. Experiments on a number of challenging synthetic and real data sets demonstrated the advantages of our approach. The experiments also reveal that our method is scalable and can successfully handle tens of cameras.

In future work, we intend to improve the the current matching algorithm by taking into account full confidence information. Also, we would like to extend the matching algorithm to deal with different variants of 3D vector fields such has higher dimensional light fields.

Acknowledgments.

This work was supported in part by an Israel Science Foundation grant 1556/10 and European Community grant PIRG05-GA-2009-248527.

References

[1] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. *IJCV*, 2007.

[2] T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. *IJCV*, 2006.

[3] S. Baker, S. Roth, D. Scharstein, M. Black, J. Lewis, and R. Szeliski. A database and evaluation methodology for optical flow. *ICCV*, 2007.

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

[5] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 1996.

[6] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for space carving. *ICCV*, 2001.

[7] T. Brox, B. Rosenhahn, D. Cremers, and H. Seidel. High accuracy optical flow serves 3D pose tracking: exploiting contour and flow based constraints. *ECCV*, 2006.

[8] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis*, 2008.

[9] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.*, 1981.

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

[11] N. Joshi, S. Avidan, W. Matusik, and D. J. Kriegman. Synthetic aperture tracking: Tracking through occlusions. 2007.

[12] N. Komodakis, G. Tziritas, and N. Paragios. Fast, Approximately Optimal Solutions for Single and Dynamic MRFs. *CVPR*, 2007.

[13] K. Kutulakos and S. M. Seitz. A theory of shape by space carving. *IJCV*, 2000.

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

[15] J. Neumann, C. Fermüller, and Y. Aloimonos. Polydioptric camera design and 3d motion estimation. In *CVPR*, 2003.

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

[17] M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. *ECCV*, 1994.

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[19] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. 1997.

[20] D. Sun, S. Roth, T. Darmstadt, and M. Black. Secrets of optical flow estimation and their principles. *CVPR*, 2010.

[21] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 1999.

[22] A. Treuille, A. Hertzmann, and S. M. Seitz. Example-Based Stereo with General BRDFs. *ECCV*, 2004.

[23] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. *ECCV*, 2008.

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

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

[26] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. *CVPR*, 2010.

[27] R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights - a progressive space carving scheme using a novel photo-consistency measure. 2003.

[28] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition (Proc. DAGM)*, 2007.