Basis Constrained 3D Scene Flow on a Dynamic Proxy

Neil Birkbeck Dana Cobzaş Martin Jägersand
University of Alberta
{birkbeck,dana,jag}@cs.ualberta.ca

Abstract

Existing scene flow approaches mainly focus on two-frame stereo-pair configurations and reconstruct an image-based representation of scene flow. Instead, we propose a variational formulation of scene flow relative to a coarse proxy geometry, which is better suited for many views. Furthermore, a linear basis is used to represent temporal surface flow, allowing for longer-range temporal correspondence with fewer variables. Our formulation takes known proxy motion into account (e.g, if the proxy is a tracked human subject), which enables 3D trajectory reconstruction when only a single view is available. Additionally, through the appropriate proxy and basis, our framework generalizes existing approaches for scene flow, optic-flow, and two-frame stereo. We illustrate results on real-data for both static and moving proxy surfaces over several frames.

1. Introduction

Image-based recovery of dense scene flow, the 3D motion of scene structure, is important for analysis of dynamic scenes. The reconstruction process often leverages temporal information to also improve the 3D structure reconstructions [22]. However, many existing approaches only consider motion between two frames (e.g., [17, 12, 20]) and do not fully utilize available coarse scene structure (or coarse scene motion through tracking).

We overcome these limitations by posing the scene flow estimation problem directly on a proxy surface. This approach generalizes the typical image-based methods (e.g., [12]), where in our formulation the proxy would be the image plane. Our formulation makes use of available proxy motion, such as coarse tracking of a human subject, allowing the reconstruction process to exploit intra-camera observations not only for motion but also for 3D structure.

To obtain longer range correspondences, we formulate the recovery of scene motion in terms of temporal basis functions. As recently demonstrated with 2D optic flow [9], a low dimensional representation uses fewer variables, provides temporal smoothness, and is less sensitive to noise.

1.1. Contributions

In our 3D formulation of basis constrained scene flow, we make the following contributions:

- Variational formulation of scene flow on a proxy surface, where the underlying proxy surface may be undergoing some known motion (e.g., from coarse tracking). This formulation naturally links many views (as opposed to two-frame, two-time methods), and directly recovers structure & flow from image intensities.
- Direct use of a linear basis in scene flow formulation that allows integration of many time steps using fewer parameters. The basis constrains and improves geometric reconstruction when the proxy surface undergoes some known motion.
- With the appropriate proxy and basis choices, our formulation generalizes and provides unified treatment of optic flow (e.g., proxy surface is the image domain), basis-constrained optical flow, two-frame stereo (e.g., no flow terms), and two frame scene flow.
- We demonstrate results on static proxies, rigidly moving proxies, and linear blend skinned proxies.

2. Related work

Proxy surfaces (e.g., triangulated base-meshes) have been used for parameterizing depth reconstructions in stereo [24, 10]. However, in scene flow, the representations are either binocular image-based [12, 20], multi-view depth map based [3], voxel-based [21], or level set based [17]; such dense methods have not been implemented on a proxy surface. The advantage of the proxy is that it already encodes an approximation of surface structure, and regularization is more natural along the surface of the proxy as opposed to the image domain.

Dense-motion capture methods for garment capture have similar goals as scene flow and make use of color-coded patterns [18], temporally align template objects to dense stereo data [5], or refine and track a dense geometry over many frames using interleaved structure and motion refinements

per-vertex [8]. Although the end goal is similar, such methods often utilize a large number of cameras, require highly textured surfaces, and don't make use of long-range temporal information when reconstructing surface motion.

The use of a temporal basis to reconstruct shape has similarities to methods that exploit temporal smoothness to aid static reconstructions. For example, scene flow can enable better per-frame geometric reconstructions [22]. Even when the correspondences between temporal geometries aren't obtained, temporal constraints improve reconstructions. For example, in the level set stereo framework, temporal regularization can be used [14], or, in the case of stereo, aggregating over a skewed window in time [25].

Subspace constraints have been successfully employed for 2D optic flow recovery. For rigid scenes, Irani used subspace constraints on the combined flow observation matrix in multi-frame flow [13]. Assuming a rigid scene and moving camera, a rank constraint on the flow matrix is derived, which helps overcome the aperture problem.

Non-rigid structure and motion techniques represent a deforming surface as a linear combination of deforming 3D shape modes [6]. Again, with a factorization method, image observations of a 3D deforming surface can be decomposed into low rank components representing the surface modes and the motion. This low rank motion basis can be extracted from reliable point tracks [19]. Once the motion basis is known, the long-range temporal motion of other (unreliable) points can be estimated simply by finding the few coefficients that make up the shape component. This empirically formed basis has been used to reconstruct dense per-pixel surface motion using a variational formulation [9], where per-pixel shape coefficients (e.g., 2-3) are estimated with regularization for long sequences (e.g., 80 frames). Per-pixel spatial basis functions have also been used for over-parameterized optic flow estimation using a variational approach [15]. We also use a variational approach, but we are interested in 3D scene flow and structure and use robust functions on the data and smoothness terms.

As in our method, simple temporal constraints (e.g., linear motion [1], constant velocity) and more expressive cosine basis (e.g., [16]) have been used to constrain the ill-posed problems of reconstructing 3D trajectories observed from moving cameras. With the exception of our previous work [4], where a constant velocity basis in a dense image-based representation, these constraints are often enforced independently per-point. In this work, we utilize these basis constraints on a moving proxy, which allows our representation to exploit this ability of structure reconstruction when there is little or no overlap in camera views.

3. Basis-constrained motion

Before describing the basis constrained scene flow on a proxy, we first use an intuitive example to illustrate the concept of using a basis to constrain temporal point motion when some coarse scene motion is known. We restrict our discussion to the case where correspondences are known, and highlight some of the benefits of using a temporal basis to represent surface flow, including the ability to overcome noise and obtain reconstructions with missing data.

Let us consider the simple case, where a single moving camera with 3×4 projection matrix, $\mathbf{P}_1$, and camera motion given by 4×4 matrices, $\{\mathbf{E}_t\}_{t=1}^{T}$, observes a dynamic scene. If correspondences of a moving point, $\mathbf{u}_t = (u_t, v_t)^\intercal$, are known, one can formulate an estimation process for the unknown moving 3D point, $\mathbf{x}(t)$, like triangulation:

$$\mathbf{x}(t) = \underset{\mathbf{x}(t)}{\operatorname{argmin}} \sum_{t=1}^{T} |\Pi(\mathbf{P}_1\mathbf{E}_t\mathbf{x}(t)) - \mathbf{u}_t|^2, \qquad (1)$$

where the symbol $\Pi(\mathbf{x})$ denotes the perspective division by z: $\Pi((x,y,z)^T) = (\frac{x}{z}, \frac{y}{z})^\intercal$. As there are 2 constraints per image, modeling the motion as independent 3D points over time (e.g., $\mathbf{x}(t) = (x_t, y_t, z_t, 1)^\intercal$) gives $3T$ unknowns, leading to an ill-posed problem. Constraining the motion, for example, by splitting the trajectory into a mean point, $\hat{\mathbf{x}}$, with a constant velocity, $\bar{o} = (o_1, o_2, o_3)$, gives fewer variables (e.g., 6):

$$\mathbf{x}(t) = \hat{\mathbf{x}} + \mathbf{o}(t) = (\hat{x}, \hat{y}, \hat{z}, 1)^\intercal + (t - t_0)(o_1, o_2, o_3, 0)^\intercal.$$

This representation allows for a unique reconstruction if $T \geq 3$ and camera motion is sufficient. The restrictive constant velocity assumption can be generalized by using a temporal basis to encode the time-varying displacements:

$$\mathbf{o}(t) = \sum_{k=1}^{H} \lambda_k \mathbf{B}_k(t). \qquad (2)$$

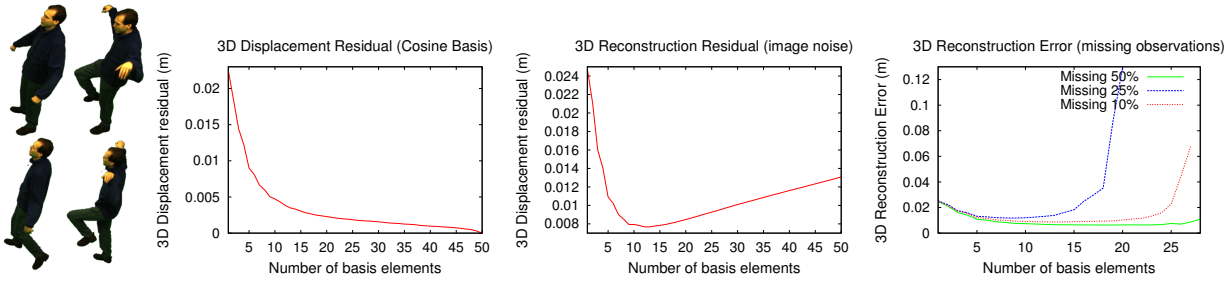
where $\{\mathbf{B}_k(t)\}_{k=1}^{H}$ are temporal basis functions ($H < T$).

This same framework holds if there are several views, $\{\mathbf{P}_i\}_{i=1}^{C}$, with correspondences $\mathbf{u}_{it}$. In this case, one can think of $\mathbf{E}_t$ as the coarse rigid motion of the scene (if known), and Eq. 1 would be accumulated over each view:

$$\mathbf{x}(t) = \underset{\mathbf{x}(t)}{\operatorname{argmin}} \sum_{i=1}^{C} \sum_{t=1}^{T} |\Pi(\mathbf{P}_i\mathbf{E}_t\mathbf{x}(t)) - \mathbf{u}_{it}|^2. \qquad (3)$$

When more than two cameras observe the point in each frame, the reconstruction is no longer ill-posed. However, using a basis to represent $\mathbf{o}(t)$ has several benefits: temporal smoothness enforced by the appropriate basis helps overcome noise, there are fewer parameters, and reconstruction is possible with missing observations.

As an example, consider a multi-view studio filming an actor. The coarse geometry can be tracked, which gives the motion of each bone over the sequence. However, the real



Motion | Displacement residual | Image-based with noise | Image-based missing data

Figure 1. Motion: Images 0, 20, 40, 50 from camera 1. Left: the *displacement residual* when approximating the 3D displacements using varying basis elements. Middle: two-view image-based reconstruction from noisy data using Eq. 4 achieves better results when using roughly 13 basis elements. Right: using 10-20 basis elements enables reconstruction with missing data from the second view.

surface exhibits residual non-rigid motion in addition to the skeleton (e.g., clothing). For a point on the surface, the coarse motion of the scene, $\mathbf{E}_t$, is given by the corresponding bone motion; this motion will be different for points attached to different bones. A linear algebraic solution for the displacement trajectory (i.e., the λ_k coefficients) can be obtained by removing the perspective division by multiplying by the z-component, giving a least squares space-time triangulation problem:

$$\begin{aligned}\lambda_k = &\underset{\lambda_k}{\operatorname{argmin}} \sum_{i,t} ([\mathbf{P}_i\mathbf{E}_t]_1 - u_{it}[\mathbf{P}_i\mathbf{E}_t]_3)(\hat{\mathbf{x}} + \sum_k \lambda_k \mathbf{B}_k(t))^2 \\ &+ \sum_{i,t} ([\mathbf{P}_i\mathbf{E}_t]_2 - v_{it}[\mathbf{P}_i\mathbf{E}_t]_3)(\hat{\mathbf{x}} + \sum_k \lambda_k \mathbf{B}_k(t))^2.\end{aligned} \tag{4}$$

where $[\mathbf{P}]_i$ is the i-th row of $\mathbf{P}$.

To illustrate this formulation, we test the ability of a low dimensional basis to approximate and reconstruct non-rigid trajectories layered on a human skeleton. The motion capture, deformed meshes, and surface displacements are from the MIT *crane* data set [23]. We approximated the displacements relative to the bone in the first 50 frames with the discrete cosine basis. Figure 1 illustrates 3D residuals for an increasing number of basis functions. At about 15 basis functions a good approximation of the surface motion is obtained; this requires only $\frac{1}{3}$ the parameters of the unconstrained motion. Furthermore, when reconstructing displacements from two views with noisy image observations using Eq. 4, fewer variables (e.g,. 12 basis functions) gives better results than more variables (middle Fig. 1), implying the lower dimensional basis helps in the presence of noise.

The ability to reconstruct in the presence of missing data is also illustrated in the right of Figure 1, where 10%, 25%, and 50% of the observations in the second camera were unknown. In these cases, solving for unconstrained motion is ill-posed, and the best reconstruction comes from restricting the motion to a lower (e.g., 10-20) dimensional basis.

4. Scene flow on a proxy

In the previous section, we demonstrated the use of a temporal basis to recover flow when coarse scene motion ($\mathbf{E}_t$) was known and correspondences were given. In this section, we replace this idea of coarse motion (e.g., motion of bones) with the formalism of a moving approximate proxy surface. Instead of assuming known correspondences, the low dimensional basis constraint is used to directly estimate the scene flow and scene structure relative to the proxy surface. Below, we formulate the problem (§4.1), give a general form of a proxy surface (§4.2), and formulate scene flow estimation relative to this proxy surface (§4.3).

4.1. Problem Definition

Given input images, $I_{i,t}$, taken from $1 \leq i \leq C$ cameras at stationary viewpoints over time $t \in \{1, \cdots, T\}$, the objective is to reconstruct the dense structure of the surface and the 3D surface flow with respect to a known approximate proxy-surface. The camera calibration is given: $\mathbf{P}_i = [\mathbf{K}_i|\mathbf{0}][\mathbf{R}_i\mathbf{t}_i]$, with internals $\mathbf{K}_i$, and external rotation, $\mathbf{R}_i$, and translation $\mathbf{t}_i$. As a shorthand, we use $\Pi_i(\mathbf{x}) = \Pi(\mathbf{P}_i\mathbf{x})$. The proxy may be moving; we assume this motion is approximately known through an external tracking process.

4.2. Proxy surface

We assume that we have available a known 3D proxy surface $\hat{\mathbf{x}}(u, v, t) : (\Omega \subset \mathbb{R}^2) \times \mathbb{Z} \mapsto \mathbb{R}^3$, embedded in 2D at discrete time steps t. We define the surface normal as

$$\mathbf{n}(u, v, t) = \frac{\hat{\mathbf{x}}_u(u, v, t) \times \hat{\mathbf{x}}_v(u, v, t)}{|\hat{\mathbf{x}}_u(u, v, t) \times \hat{\mathbf{x}}_v(u, v, t)|}, \tag{5}$$

where $\mathbf{x}_u$ and $\mathbf{x}_v$ are the partial derivatives of the surface (i.e., tangents). The tangent frame matrix is defined as

$$\mathbf{T}(u, v, t) = \left[\frac{\hat{\mathbf{x}}_u(u, v, t)}{|\hat{\mathbf{x}}_u(u, v, t)|}, \frac{\hat{\mathbf{x}}_v(u, v, t)}{|\hat{\mathbf{x}}_v(u, v, t)|}, \mathbf{n}(u, v, t)\right]. \tag{6}$$

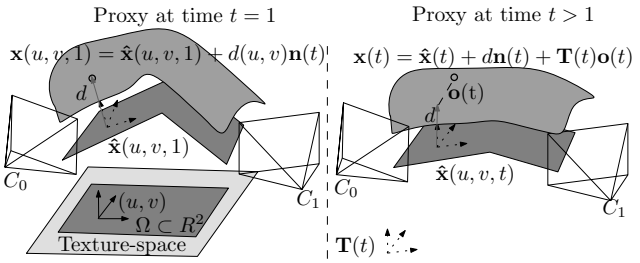


Figure 2. We represent the surface relative to a possibly moving proxy surface observed by several cameras. At a reference frame (e.g., $t = 1$), the true surface is simply a displacement d from the proxy along the normal. In subsequent frames, the surface is represented as the displaced point with an additive flow component.

4.3. Recovering displacements and flow

Scene motion is modeled relative to the moving scene proxy with a time dependent 3D temporal offset, $\mathbf{o}(u,v,t)$, and a displacement along the normal $d(u,v)$:

$$\mathbf{x}(u,v,t) = \hat{\mathbf{x}} + \mathbf{T}(u,v,t)\mathbf{o}(u,v,t) + d(u,v)\mathbf{n}(u,v,t). \quad (7)$$

The displacement, d, is with respect to a reference time, e.g., $t = 1$. The third component of the offset vector $\mathbf{o}(u,v,t) = (o_1(u,v,t), o_2(u,v,t), o_3(u,v,t))^\intercal$, accounts for temporal changes in normal motion.[1] Using, $\mathbf{T}$, the tangent frame of the surface, implies that the offsets are represented in the coordinate frame of the surface (see Figure 2). Setting $\mathbf{T}$ to the identity means the surface motion would be represented in the global world coordinate frame.

Unlike the previous section, which assumed known correspondences, here we recover 3D surface displacement, d, and long range flow, $\mathbf{o}$, directly from image intensities. The desired displacements and flow should be both photo-consistent and flow-consistent. Therefore, the displacement and flow can be recovered by minimizing the following functional, which uses brightness constancy to measure flow- and photo-consistency:

$$F(d,\mathbf{o}) = \alpha \sum_{t=2}^{T} \underbrace{F_f(d,\mathbf{o},t)}_{Flow} + \sum_{t=1}^{T} (\underbrace{F_s(d,\mathbf{o},t)}_{Stereo} + \underbrace{F_r(d,\mathbf{o},t)}_{Regularize}), \quad (8)$$

The individual terms are

$$F_s(d,\mathbf{o},t) = \int_\Omega \sum_{j \neq i} w_{ijt} \Psi(|I_{it}(\Pi_i(\mathbf{x}(t))) - I_{jt}(\Pi_j(\mathbf{x}(t)))|^2) dA$$

$$F_f(d,\mathbf{o},t) = \int_\Omega w_{it1} \Psi(|I_{i,t}(\Pi_i(\mathbf{x}(t))) - I_{i,1}(\Pi_i(\mathbf{x}(1)))|^2) dA$$

$$F_r(d,\mathbf{o},t) = \beta_2 \int_\Omega \Psi(|\nabla \mathbf{o}(u,v,t)|^2) dA + \beta_1 \int_\Omega \Psi(|\nabla d|^2) dA$$

[1] Due to o_3, the displacement component, d, can be dropped. However, we find it advantageous to regularize the displacement component separately, and instead keep d and choose basis functions with zero displacement at the reference time.

$$|\nabla \mathbf{o}|^2 = \sum_{j=1}^{3} |\nabla o_j|^2.$$

where $\mathbf{x}(t)$ is a shorthand for $\mathbf{x}(u,v,t)$ in Eq. 7, $dA = dudv$, and $\Psi(x^2) = \sqrt{x^2 + \epsilon^2}$ is a smooth L1 norm [7], and ∇ is a 2D spatial gradient over the parameters u and v. The weighting term $w_{ijt}(u,v)$ (resp. w_{it1}) is used to encode visibility or reliability of the image observations for stereo pairs (resp. flow). See Section 5.3 for details.

The stereo term, F_s, ensures the displaced surface is photo-consistent at each time; the flow term, F_f, like optic flow, ensures the intensity at the flowed points is in agreement with the reference frame (e.g., $t = 1$). The regularization acts both on the displacements d and prefers to have smoothly varying 3D flow at each time t.

4.4. Temporal basis-constrained flow

The functional, F, is dependent on the shape displacement d, and the surface offsets $\mathbf{o}(u,v,t)$, at each time $2 \leq t \leq T$. Allowing arbitrary displacements would require a large number of variables and would need an extra term for temporal smoothness. Instead, as in Eq. 2, the motion is modeled with a low-dimensional basis, but the coefficients of motion, λ_k, are spatially varying scalar fields:

$$\mathbf{o}(u,v,t) = \sum_{k=1}^{H} \mathbf{B}_k(t) \lambda_k(u,v). \quad (9)$$

Where again $\{\mathbf{B}_k(t)\}_{k=1}^H$ are a set of basis functions. For example, a constant velocity basis using time 1 as a reference would use $\mathbf{B}_1(t) = (t-1, 0, 0)^\intercal$, $\mathbf{B}_2(t) = (0, t-1, 0)^\intercal$ and $\mathbf{B}_3(t) = (0, 0, t-1)^\intercal$. The total number of unknown scalar fields reduces from $3(T-1)+1$ to $H+1$.

The basis elements can be more elaborate. As in the 2D flow (e.g., [9]), the motion basis functions can be factored from a sparse set of representative tracks. We utilize the discrete cosine basis, which has been used to constrain trajectories of moving points observed from moving cameras [16]. As a consequence, when proxy motion is known (and sufficient), it is possible to reconstruct the 3D flow trajectory when the surface is only visible by one camera.

An advantage of this framework is that it naturally generalizes a number of variational approaches involving dense correspondence, including optic flow, stereo, and scene flow. Table 1 lists the corresponding proxy and basis functions for a number of these applications.

4.5. Euler-Lagrange equations

Scalar fields d and $\{\lambda_k\}_{k=1}^H$ that minimize Eq. 8 must satisfy the Euler-Lagrange equations:

Problem	Time	Proxy (static/dynamic)	An example basis
Optic flow [7]	$t = 2$	Image plane (static)	$\mathbf{B}_1(t) = [1, 0, 0], \mathbf{B}_2(t) = [0, 1, 0]$
Optic flow with basis [9]	$t > 1$	Image plane (static)	E.g., $\mathbf{B}_1(t) = [t, 0, 0], \mathbf{B}_2(t) = [0, t, 0]$
Stereo with basis for depth	$t \geq 1$	Image plane (static)	E.g., $\mathbf{B}_1 = [0, 0, t]$, cosine basis, ...
Displacement from proxy (e.g., [24])	$t \geq 1$	Approx geom. (dynamic)	E.g., $\mathbf{B}_1 = [0, 0, t]$, cosine basis, ...
Scene flow [12]	$t = 2$	Image plane (static)	$\mathbf{B}_1 = [1, 0, 0], \mathbf{B}_2 = [0, 1, 0], \mathbf{B}_2 = [0, 0, 1]$
Scene flow on proxy	$t > 1$	Approx geom. (dynamic)	$\mathbf{B}_1 = [t, 0, 0], \mathbf{B}_2 = [0, t, 0], \mathbf{B}_2 = [0, 0, t]$

Table 1. Our basis-constrained scene flow on a proxy generalizes a number of approaches. In optic flow, simple stereo, or scene flow the proxy would be the image plane. The basis functions limit the reconstruction to the 2D offsets (for optic flow) or the depth (for stereo), and can easily extend to multi-frame estimates. Our application is the most general: scene flow on top of a dynamic proxy.

$$0 = \sum_{t=1}^{T} \sum_{j \neq i} w_{ijt} \Psi'(I_{ijt}^2) I_{ijt} \frac{\partial}{\partial d} I_{ijt} + \alpha \sum_{t=2}^{T} w_{it1} \Psi'(I_{it1}^2) I_{it1} \frac{\partial}{\partial d} I_{it1} - \beta_1 \nabla \cdot (\Psi'(|\nabla d|^2) \nabla d), \tag{10}$$

and for all $1 \leq k \leq H$:

$$\sum_{t=1}^{T} \sum_{j \neq i} w_{ijt} \Psi'(I_{ijt}^2) I_{ijt} \frac{\partial}{\partial \lambda_k} I_{ijt} + \alpha \sum_{t=2}^{T} w_{it1} \Psi'(I_{it1}^2) I_{it1} \frac{\partial}{\partial \lambda_k} I_{it1} - \beta_2 \nabla \cdot (\Psi'(|\nabla \mathbf{o}|^2) \sum_{c=1}^{3} \nabla o_c [\mathbf{B}_k(t)]_c) = 0. \tag{11}$$

Here, the shorthand $I_{ijt} = I_{i,t}(\Pi_i(\mathbf{x})) - I_{j,t}(\Pi_j(\mathbf{x}))$, $I_{it1} = I_{i,t}(\Pi_i(\mathbf{x})) - I_{i,1}(\Pi_i(\mathbf{x}))$. The partial derivatives of these terms w.r.t d and λ_k are

$$\frac{\partial}{\partial d} I_{ijt} = \nabla I_{i,t} \frac{\partial}{\partial d} \Pi_i(\mathbf{x}) - \nabla I_{j,t} \frac{\partial}{\partial d} \Pi_j(\mathbf{x}), \tag{12}$$

$$\frac{\partial}{\partial d} \Pi_i(\mathbf{x}) = \underbrace{\Pi'(\mathbf{K}_i(\mathbf{R}_i \mathbf{x} + \mathbf{t}_i))}_{2 \times 3} \underbrace{\mathbf{K}_i \mathbf{R}_i \mathbf{n}(u, v, t)}_{3 \times 1}, \tag{13}$$

$$\frac{\partial}{\partial \lambda_k} I_{ijt} = \nabla I_{i,t} \frac{\partial}{\partial \lambda_k} \Pi_i(\mathbf{x}) - \nabla I_{j,t} \frac{\partial}{\partial \lambda_k} \Pi_j(\mathbf{x}), \tag{14}$$

$$\frac{\partial}{\partial \lambda_k} \Pi_i(\mathbf{x}) = \Pi'(\mathbf{K}_i(\mathbf{R}_i \mathbf{x} + \mathbf{t}_i)) \mathbf{K}_i \mathbf{R}_i \mathbf{T}(u, v, t) \mathbf{B}_k(t), \tag{15}$$

$$\Pi'((x\ y\ z)^\intercal) = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix}. \tag{16}$$

$\frac{\partial}{\partial d} I_{it1}$ and $\frac{\partial}{\partial \lambda_k} I_{it1}$ are similar.

The above Euler-Lagrange equations are linearized and solved using multi-grid methods (see Appendix A).

5. Implementation on a Skinned Proxy Mesh

The formulation in the previous section operated on a generic, possibly moving proxy surface. In this section, we specify implementation details for a linear-blend skinned triangulated proxy mesh (§5.1), how the tangent space is defined (§5.2), and further describe the initialization and optimization (§5.3).

5.1. Skinned mesh

A skinned mesh consists of a set of vertices, $\hat{V} = \{\hat{\mathbf{v}}_j\}$, in a rest-pose attached to one or more skeleton bones, and a set of triangles $\mathcal{S} = \{s_i \in \mathbf{Z}^3\}$. Each bone, b, has a transformation, $\mathbf{A}_b(\theta_b)$, relative to its parent, $p(b)$; the concatenation up the chain gives the transformation from bone to world: $\mathbf{M}_{b,j}(\theta) = \mathbf{M}_{p(b),j}(\theta) \mathbf{A}_b(\theta_b)$. Given a set of angles, $\theta = \bigcup_b \theta_b$, representing the pose of the skeleton, the deformed vertex is given as

$$\mathbf{v}_j(\theta) = \sum_b w_{b,j} \mathbf{M}_{b,j}(\theta) \mathbf{M}_{b,j}(\mathbf{0})^{-1} \hat{\mathbf{v}}_j, \tag{17}$$

where $w_{b,j}$ is the attachment weight of vertex j to bone b. The motion of the proxy mesh is then determined by the temporal sequence of joint angles $\{\theta^t\}_{t=1}^T$.

5.2. Tangent space

Each triangle s_i has a corresponding set of 2D texture coordinates, $(\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \mathbf{u}_{i,3})$. The piecewise linear mapping from 2D to 3D for a point $(u_i, v_i)^T$ in triangle $(\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \mathbf{u}_{i,3})$ with bary-centric coordinates $(u_i, v_i)^T = \sum_{j=1}^3 a_j \mathbf{u}_{i,j}$, is

$$\hat{\mathbf{x}}(u, v, t) = \sum_{j=1}^{3} a_j \mathbf{v}_{s_{i,j}}. \tag{18}$$

The tangent space is also interpolated across triangles in the 2D domain using these same bary-centric weights, a_j. The surface normal at each vertex is the average of unnormalized triangle normal (e.g., $(\mathbf{v}_{\mathbf{s_{i,2}}} - \mathbf{v}_{\mathbf{s_{i,1}}}) \times (\mathbf{v}_{\mathbf{s_{i,3}}} - \mathbf{v}_{\mathbf{s_{i,1}}})$).

Tangent directions, for each triangle are approximated for each triangle

$$\mathbf{x}_x = \frac{[\mathbf{u}_{i,3} - \mathbf{u}_{i,1}]_y (\mathbf{v}_{i,2} - \mathbf{v}_{i,1}) - [\mathbf{u}_{i,2} - \mathbf{u}_{i,1}]_y (\mathbf{v}_{i,3} - \mathbf{v}_{i,1})}{c}$$

$$\mathbf{x}_y = \frac{[\mathbf{u}_{i,2} - \mathbf{u}_{i,1}]_x (\mathbf{v}_{i,3} - \mathbf{v}_{i,1}) - [\mathbf{u}_{i,3} - \mathbf{u}_{i,1}]_x (\mathbf{v}_{i,2} - \mathbf{v}_{i,1})}{c}$$

$$c = [\mathbf{u}_{i,2} - \mathbf{u}_{i,1}]_x [\mathbf{u}_{i,3} - \mathbf{u}_{i,1}]_y - [\mathbf{u}_{i,3} - \mathbf{u}_{i,1}]_x [\mathbf{u}_{i,2} - \mathbf{u}_{i,1}]_y.$$

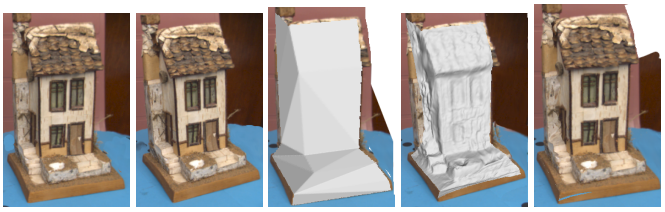


Figure 3. From left: two of four input views, the initial proxy, the recovered displaced mesh, and the textured displaced mesh.

As in the case for normals, the tangents for each vertex are averaged over the triangles that contain it.

5.3. Initialization and Optimization

The proxy mesh can either come from a sparse set of stereo points or a manually initialized selection of points. The uv-parameterization can be obtained by projection of vertices into a single-view, through automatic methods, or it can be aided by a user in modeling software.

In the case of no approximate proxy motion, all motion will be relative to the original proxy surface. When a kinematic skeleton is available, we track the surface of the skeleton using silhouettes or image-based scores.

The image weights, w_{ijt}, are set as $w_{ijt}(u,v) = w_{it}(u,v)w_{jt}(u,v)$, where

$$w_{it}(u,v) = \begin{cases} \langle \mathbf{n}(u,v,t), \mathbf{l}_i \rangle & \text{if } \mathbf{x}(u,v,t) \text{ visible in } I_i \\ 0 & \text{otherwise,} \end{cases}$$

and $\mathbf{l}_i$ is the ray to camera i. The flow weights, w_{it1}, are defined similarly.

The Euler-Lagrange equations (Eq. 10 & 11) are solved on multiple resolutions of the tangent space and the input images. Solutions from lower resolutions are propagated to the higher resolutions by simple image resampling.

6. Experiments

We now illustrate the results of our implementation on several synthetic and real data sets.

Static proxy, depth only Figure 3 illustrates one of the applications that our framework generalizes: depth from a static proxy. Four views of the toy house (one time frame) are used to recover depth from a simple manually created proxy (no temporal basis are used).

Skinned object, no camera overlap In this synthetic data set we illustrate the helpful effect of using the basis constraints in the case of kinematic motion when there is no overlap between the camera views. With this experiment we show that constraining the motion of the surface with a basis can improve the surface reconstruction. The sequence contains 3 views and 3 time instances of a cylindrical object (roughly 3 units high by 1.5 units wide) with two bones undergoing a small bend and small translation motion (see

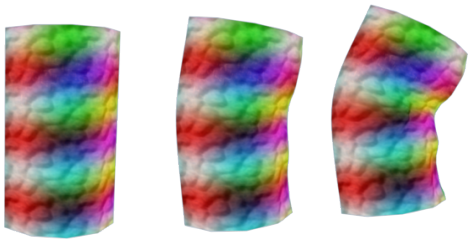


Figure 4. The three input images (from camera 1) for the synthetic skinned cylinder illustrate the motion and texture on the object.

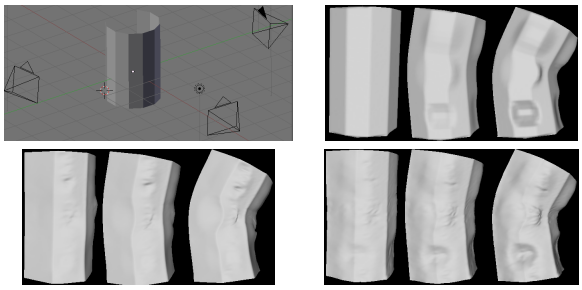


Figure 5. Top left: the non-overlapping camera configuration. Top right: ground truth geometry from first viewpoint. Bottom left: inaccurate reconstruction results with no constraints. Bottom right: more accurate reconstruction using a constant velocity constraint.

Fig. 4 for input and Fig. 5 for camera configuration). The base motion of the skeleton is assumed to be known, and the ground truth skinning weights are used. The underlying motion over the three frames is linear.

We reconstructed the surface motion both with a constant velocity basis and with an unconstrained flow. The use of a constant velocity basis allows more accurate reconstruction of the depth (Fig. 5). Although the unconstrained basis does recover some of the tangential flow, the depth estimates are still inaccurate. The average of the median position error at each time step was 0.0209 and 0.0135 for the unconstrained and constrained reconstructions respectively (base surface error, with no displacements, is 0.0722).

Static proxy, multiple views In this example, we illustrate that the proxy motion need not be known. The data was collected from a static stereo setup that captured a translating deforming mouse pad (Fig. 6). The base geometry consisted of 2 triangles specified manually in the first frame. We used a cosine basis with 8 basis elements (for each flow component) to model the motion over 16 frames. The motion was successfully modeled (illustrated by the reprojection of texture) and the flowed geometric surface exhibits the deformations of the mouse pad.

Rigid moving proxy, single view In this example, we illustrate that taking motion of the proxy into account aids the reconstruction. We used a single view from the previous experiment, and used the flow results from the multi-view case to fit the rigid motion of the proxy. The time-varying depth

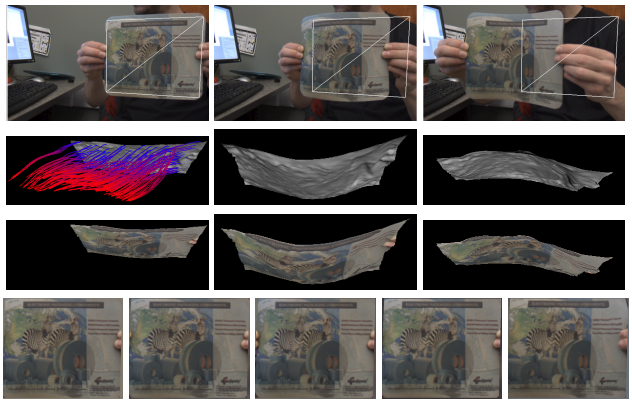


Figure 6. Top: input frames 0, 7, and 15 from the mousepad sequence (white rectangle shows static proxy). Middle: shaded and textured reconstructions (several trajectories plotted in $t = 0$). Bottom: rectified surface texture over several frames.

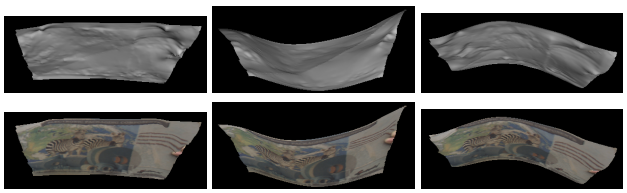


Figure 7. Shaded and texture reconstructions for the single-view time-varying depth only reconstruction of the mousepad on frames 0, 7, and 15.

Figure 8. Top: input images from time 0, 5, 14. Middle: reconstructed geometry. Bottom: deformed textured with $I_{1,0}$.

on the surface was then recovered over the sequence using the first 3 cosine basis functions to model the displacement along the normal (flow components were not modeled). As illustrated in Figure 7, the time-varying depth was successfully recovered from the single-view sequence.

Two link arm, skinned proxy In this example we demonstrate the use of the full model: a moving skinned proxy observed by multiple cameras. The proxy object is a moving arm (Fig. 8). Two views were used to track the geometry and to obtain an initial shape (a skeleton was manually inserted and skinning weights were obtained automatically [2]). The texture resolution was 128x128 and a cosine basis with 7 elements for each coordinate were used to model the motion over 16 frames. Notice that the geometry starts to recover the wrinkles.

7. Discussion & Conclusion

We have presented a variational approach for scene flow (and structure) reconstruction directly from image intensities that uses a temporal basis for longer range tracks. The formulation takes advantage of known motion of the proxy (when available) to enable monocular reconstruction.

Although the proxy representation is natural for representing structure and flow, it does have limitations. One such limitation is that the proxy must be appropriate for the scene (e.g., the underlying unknown scene must be a function of depth from the proxy). Further, as only brightness constraints are used, the surface flow can be confused when there are illumination changes (likely due to relative surface motion w.r.t. the lights). This can be addressed by using a data term less sensitive to illumination (e.g., using the gradient of the image).

The linearization (Appendix A) uses $(H+1) \times (H+1)$ tensors for each data term (and each pixel). This limits the number of basis functions that can be used for long range sequences. Removing the robust norms on data terms may allow for a more efficient memory implementation.

Furthermore, our implementation currently doesn't handle discontinuities in the uv-parameterization, although this is possible in the discretization (e.g., as in [11]). For other future work, we would like to consider coupling the tracking and flow into a single formulation; in this way, the tracking module could take advantage of the refined surface.

Also, the decision of the type of motion bases (and the degree) are currently parameters that must be chosen by the user. If possible, prior knowledge of the flow should be used to guide these decisions. However, in future work, we would like to ease this decision by making use of an empirically learned motion basis from a set of reliable point tracks (e.g., as in the 2D flow implementations [9]).

A. Solving the Euler-Lagrange

Following optic flow [7], the Euler-Lagrange equations (Eqs 10 & 11) are solved with fixed point iterations:

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{j \neq i} w_{ijt} \Psi'_{ij}((I^{h+1}_{ijt})^2) I^{h+1}_{ijt} \frac{\partial}{\partial d} I^{h}_{ijt} + \\ &\alpha \sum_{t=2}^{T} w_{it1} \Psi'((I^{h+1}_{it1})^2) I^{h+1}_{it1} \frac{\partial}{\partial d} I^{h}_{it1} - \\ &\beta_1 \nabla \cdot (\Psi'(|\nabla d^{h+1}|^2) \nabla d^{h+1}) = 0, \end{aligned} \tag{19}$$

$$\sum_{t=1}^{T}\sum_{j\neq i} w_{ijt}\Psi'_{ij}((I^{h+1}_{ijt})^2)I^{h+1}_{ijt}\frac{\partial}{\partial\lambda_k}I^h_{ijt}+$$
$$\alpha\sum_{t=2}^{T} w_{it1}\Psi'((I^{h+1}_{it1})^2)I^{h+1}_{it1}\frac{\partial}{\partial\lambda_k}I^h_{it1}- \quad (20)$$
$$\beta_2\nabla\cdot(\Psi'(|\nabla\mathbf{o}^{h+1}|^2)\sum_{c=1}^{3}\nabla o^{h+1}_j[\mathbf{B}_k(t)]_j)=0.$$

Defining the update, $d^{h+1}=d^h+\delta d^h$, $\lambda^{h+1}_k=\lambda^h_k+\delta\lambda^h_k$, $1\leq k\leq H$, the data terms can now be linearized:

$$\begin{aligned} I^{h+1}_{ijt} =& I_{i,t}(\Pi_i(\mathbf{x}^{h+1}))-I_{j,t}(\Pi_j(\mathbf{x}^{h+1}))\\ \approx& I_{i,t}(\Pi_i(\mathbf{x}^h))-I_{j,t}(\Pi_j(\mathbf{x}^h))+\\ & (I^h_{ijt})_d\delta d^h+\sum_k(I^h_{ijt})_{\lambda_k}\delta\lambda_k=(\nabla I^h_{ijt})^\intercal\delta\mathbf{D}^h. \end{aligned} \quad (21)$$

Where

$$\nabla I^h_{ijt}=[(I^h_{ijt})_d,(I^h_{ijt})_{\lambda_1},\cdots,(I^h_{ijt})_{\lambda_H},I^h_{ijt}]^\intercal, \quad (22)$$
$$\delta\mathbf{D}^h=[\delta d^h_k,\delta\lambda^h_1,\cdots,\delta\lambda^h_H]^\intercal. \quad (23)$$

The terms for I_{it1} are defined analogously. Defining the data tensors $S^h_{ijt}=(\nabla I^h_{ijt})(\nabla I^h_{ijt})^\intercal$ and $S^h_{it1}=(\nabla I^h_{it1})(\nabla I^h_{it1})^\intercal$ the linearized equations are

$$\sum_{t=1}^{T}\sum_{j\neq i} w_{ijt}\Psi'_{ijt}(\cdot)[S_{ijt}]_1\delta\mathbf{D}^h+\alpha\sum_{t=2}^{T} w_{it1}\Psi'_{it1}(\cdot)[S_{ijt}]_1\delta\mathbf{D}^h$$
$$-\beta_1\nabla\cdot(\Psi'(|\nabla d^{h+1}|^2)\nabla d^{h+1})=0,$$
$$\sum_{t=1}^{T}\sum_{j\neq i} w_{ijt}\Psi'_{ijt}(\cdot)[S_{ijt}]_{k+1}\delta\mathbf{D}^h+$$
$$\alpha\sum_{t=2}^{T} w_{it1}\Psi'_{it1}(\cdot)[S_{ijt}]_{k+1}\delta\mathbf{D}^h-$$
$$\beta_2\nabla\cdot(\Psi'(|\nabla\mathbf{o}^{h+1}|^2)\sum_{c=1}^{3}\nabla o^{h+1}_j[\mathbf{B}_k(t)]_j)=0,$$
$$\Psi'_{ij*}(\cdot)=\Psi'((\delta\mathbf{D}^h)^\intercal S_{ij*}\delta\mathbf{D}^h).$$

The discretization and multi-grid optimization follow the framework for optic flow [7].

References

[1] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*, 22(4):348–357, 2000.

[2] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.

[3] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. In *CVPR '10*, pages 1506–1513, 2010.

[4] N. Birkbeck, D. Cobzas, and M. Jagersand. Monocular depth and scene flow under constant velocity. In *3DPVT*, 2010.

[5] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *ACM SIGGRAPH 2008 Papers*, 2008.

[6] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR '00*, volume 2, pages 690–696, 2000.

[7] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV '05*, pages 749–755, 2005.

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

[9] R. Garg, L. Pizarro, L. Agapito, and D. R uckert. Dense multi-frame optic flow for non-rigid objects using subspace constraints. In *ACCV '10*, pages 460–473, 2010.

[10] B. Goldlücke and D. Cremers. A superresolution framework for high-accuracy multiview reconstruction. In *31st DAGM Symposium on Pattern Recognition*, pages 342–351, 2009.

[11] B. Goldlücke and D. Cremers. Superresolution texture maps for multiview reconstruction. In *ICCV '09*, pages 1677–1684, 2009.

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

[13] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV '99*, pages 626–633, 1999.

[14] M. Magnor and B. Goldlucke. Spacetime-coherent geometry reconstruction from multiple video streams. In *3DPVT '04*, pages 365–372, 2004.

[15] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *Int. J. Comput. Vision*, 76:205–216, February 2008.

[16] H. S. Park, T. Shiratori, I. Matthews, and Y. A. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *ECCV '10*, September 2010.

[17] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *CVPR '05*, pages 822–827, 2005.

[18] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. *Computer Graphics Forum*, 24(3):439–448, Aug. 2005.

[19] L. Torresani and C. Bregler. Space-time tracking. In *ECCV '02*, pages 801–812, 2002.

[20] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *ECCV'10*, pages 568–581, 2010.

[21] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, 27(3):475–480, 2005.

[22] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6d. In *CVPR '00*, June 2000.

[23] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):1–9, 2008.

[24] G. Vogiatzis, P. Torr, S. Seitz, and R. Cipolla. Reconstructing relief surfaces. In *BMVC '04*, pages 117–126, 2004.

[25] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *CVPR '03*, pages 367–374, June 2003.