Multi-scale 3D Scene Flow from Binocular Stereo Sequences

Rui Li and Stan Sclaroff *
Computer Science Department
Boston University
Boston, MA 02215

Abstract

Scene flow methods estimate the three-dimensional motion field for points in the world, using multi-camera video data. Such methods combine multi-view reconstruction with motion estimation approaches. This paper describes an alternative formulation for dense scene flow estimation that provides convincing results using only two cameras by fusing stereo and optical flow estimation into a single coherent framework. To handle the aperture problems inherent in the estimation task, a multi-scale method along with a novel adaptive smoothing technique is used to gain a regularized solution. This combined approach both preserves discontinuities and prevents over-regularization – two problems commonly associated with basic multi-scale approaches. Internally, the framework generates probability distributions for optical flow and disparity. Taking into account the uncertainty in the intermediate stages allows for more reliable estimation of the 3D scene flow than standard stereo and optical flow methods allow. Experiments with synthetic and real test data demonstrate the effectiveness of the approach.

1. Introduction

Over the past years, there has been increasing interest in methods that can estimate the motion of a 3D scene given video streams obtained via a multi-camera rig. While the demonstrated applications of the estimation of non-rigid 3D motion are impressive, a number of aspects of the estimation of 3D motion problem remain open. In particular, the estimation of 3D motion is generally susceptible to noise when a small number of cameras is used in the stereo-rig. There are also problems with estimation errors in regions of low contrast variation, or in regions where the surface is visible in only a subset of the views.

In this paper, we propose an improved algorithm for the computation of nonrigid 3D scene flow [19, 23], given only *binocular* video streams. Three-dimensional scene flow represents scene motion in terms of a dense 3D vector field, defined over every visible surface in the scene. We present a multi-scale estimation framework that quantifies and accounts for the estimation errors of 3D scene flow that arise in regions of low contrast variation. The framework extends the basic notions of multi-scale distributions of optical flow [15] to 3D scene flow, and it employs a region-based method in order to gain a reliable solution. This combined approach both preserves discontinuities and prevents over-regularization – two problems commonly associated with basic multi-scale approaches. The improved framework yields good results for the binocular case and can be easily extended to the multi-baseline case.

[1]This research was funded in part by NSF grants CNS-0202067 and IIS-0208876, and ONR N00014-03-1-0108.

2. Related Work

There has been a fairly large amount of research done in the area of 3D motion estimation. We broadly classify the related work into four categories based on the setup and assumptions made.

Rigid motion, monocular sequence: Structure-from-motion techniques [16] recover relative motion together with scene structure from a monocular image sequence. The scene is generally assumed to be rigid [16] or piecewise rigid [4]; thus, only a restricted form of non-rigid motion can be analyzed via these techniques [1].

Non-rigid motion, monocular sequence: By making use of strong *a priori* knowledge, or by directly modelling assumptions about the scene, techniques like [11, 12, 17] can estimate non-rigid motion from a monocular image sequence. The method of [17] assumes that the motion minimizes the deviation from a rigid body motion. In other approaches [11, 12], a deformable model is used and the 3D motion is recovered by estimating the parameters to deform a predefined model.

Motion stereo: With multiple cameras, stereo and 2D motion information can be combined to recover the 3D motion, *e.g.,* [8, 10, 13, 20, 21, 25]. Except for [8] and [10], almost all techniques in this category assume rigid motion. For non-rigid tracking, [8] uses relaxation-based algorithms and [10] generalizes the model-based approach of [11]. The first approach cannot provide dense 3D motion while the

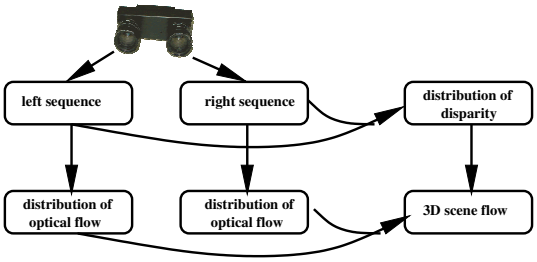

Figure 1. *System Overview*

latter approach needs *a priori* knowledge of the scene, *i.e.,* the deformable model.

Non-rigid motion, multi-view: Vedula, *et al.*, [19] introduce the concept of dense scene flow as the 3D counterpart of optical flow. They present a linear algorithm to compute scene flow from multi-view optical flow. Given scene flow and initial 3D scene structure, dynamic scene structure can be recovered. Zhang *et al.,* [23] reformulated the scene flow estimation problem in terms of energy minimization; scene flow is computed by fitting an affine motion model to image partitions with global smoothness constraints. This algorithm was further improved [24] so that discontinuities are preserved. Two other approaches [5, 22] recover shape from dynamic scenes by finding correspondence in a 3D space time window. Structured light is used to improve the accuracy in correspondence matching. However, [5, 22] require active illumination and do not estimate inter-frame motion.

Our method is in this final category, in particular it improves upon Vedula's [19].

3. Overview

Our system is presented in Figure 1. The binocular camera rig is calibrated and the images captured are rectified. Both distributions of optical flow and disparity are computed based on the unified approach described in Sections 4.1 and 4.2. 3D scene flow is then computed by combining the distribution information of flow and disparity together as described in Section 4.3.

4. Approach

Both optical flow and disparity can be formulated as problems of finding corresponding points in two images. Optical flow finds correspondence in time while disparity finds correspondence in different views. Let $f(x_i, y_i, c)$ be the function of position and time/view for the image signal ($c = t$ for time, $c = v$ for view), $\mathbf{v}$ be the pixel displacement caused by change in time or view. Commonly, the goal is to find $\mathbf{v}$ such that

$$E(\mathbf{v}) = \sum_i (\nabla \mathbf{f}_i \cdot \mathbf{v} + f_{ic})^2 \tag{1}$$

is minimized, where i is the index for pixels in the image. In the following derivations, i is dropped for simplicity. $\nabla \mathbf{f}$ represents the spatial gradient of the image and f_c represents the change in image caused either by time or view. This error function has been used both in the context of optical flow [7] and stereo vision [9]. This minimization problem is under-constrained, and thus some form of regularization is needed.

In the context of optical flow computation, Eq. 1 enforces the Constant Brightness Assumption. Usually this assumption is violated when there is a large motion between two images captured at two consecutive time steps. To alleviate this problem, multi-resolution based approaches are widely adopted. Eero Simoncelli [15] proposed an approach that computes distributions of optical flow using an image pyramid. This approach is elegant and has many potential applications, such as probabilistic tracking and motion analysis. In this paper, we adapt this approach in an improved formulation. The proposed approach takes care of the problem of over-smoothing of [15] and preserves the nice property of producing a distribution of motion estimation. The same approach is extended to estimate disparity distributions. Given the distributions of optical flow and disparity, we compute 3D scene flow via an integrated algorithm using weighted least squares described in Section 4.3.

4.1. Distributions of Flow

Following [15], the uncertainty in optical flow computation is described through the use of Gaussian noise model,

$$\nabla \mathbf{f} \cdot (\mathbf{v} - \mathbf{n}_1) + f_t = n_2. \tag{2}$$

The image intensity signal is represented as a function f of position (denoted by image coordinates x and y) and time (denoted by t). The image gradient is $\nabla \mathbf{f} = (f_x(x, y, t), f_y(x, y, t))^T$ and the temporal derivative of the image is f_t. The first random variable $\mathbf{n}_1$, modelled as $\mathbf{n}_1 \sim \mathcal{N}(0, \mathbf{\Lambda}_1)$, describes the error resulting from a failure of the planarity assumption. The second random variable, $n_2 \sim \mathcal{N}(0, \Lambda_2)$ describes the errors in the temporal derivative measurements. For the prior distribution of $\mathbf{v}$, a zero-mean Gaussian distribution with a small covariance Λ_p is used. If there is no intensity variation in the image or part of the image, Λ_p makes Eq. 2 well-conditioned.

Assume that $\mathbf{v}$ is constant in a small region, let n be the number of pixels within the neighborhood, each optical flow vector (per pixel) is considered as a normal distribution with mean flow $\hat{\mathbf{v}}$ and covariance $\Lambda_{\mathbf{v}}$ defined as follows:

$$\Lambda_{\mathbf{v}} = \left[\sum_i^n \frac{w_i \mathbf{M}_i}{\sigma_1 \|\nabla f(x_i, y_i, t)\|^2 + \sigma_2} + \Lambda_p^{-1} \right]^{-1}, \tag{3}$$

$$\hat{\mathbf{v}} = -\Lambda_{\mathbf{v}} \cdot \sum_i \frac{w_i \mathbf{b}_i}{\sigma_1 \|\nabla f(x_i, y_i, t)\|^2 + \sigma_2}, \tag{4}$$

where

$$\mathbf{M} = \nabla\mathbf{f}\nabla\mathbf{f}^T = \begin{pmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{pmatrix},\ \mathbf{b} = \begin{pmatrix} f_x f_t \\ f_y f_t \end{pmatrix},$$

w_i is the weight assigned to the neighboring pixel i, $\sigma_1\mathbf{I} = \mathbf{\Lambda}_1$ and $\sigma_2 = \Lambda_2$.

4.1.1 Coarse-to-fine Estimation of Flow Distribution

To propagate the uncertainty model at coarser scale levels (lower resolution images) to finer scale levels (higher resolution images), Simoncelli developed a filter-based coarse-to-fine algorithm [14]. We only describe the basic solution here.

Define a *state evolution* equation for the estimated flow field $\hat{\mathbf{v}}$,

$$\hat{\mathbf{v}}(l) = \mathbf{E}(l-1)\hat{\mathbf{v}}(l-1) + \mathbf{n}_0,\ \ \mathbf{\Lambda}_0 \sim \mathcal{N}(0, \mathbf{\Lambda}_0), \quad (5)$$

where l is an index for scale (larger values of l correspond to finer scale). $\mathbf{E}$ is a linear interpolation operator used to extend a coarse scale flow field to finer scale. The random variable $\mathbf{n}_0$ represents the uncertainty of the prediction of the finer-scale flow field from the coarser-scale flow field, it is assumed to be point-wise independent, zero-mean and normally distributed.

The measurement equation is defined based on Eq. 2:

$$-f_t(l) = \nabla\mathbf{f}(l)\cdot\mathbf{v}(l) + (n_2 + \nabla\mathbf{f}(l)\cdot\mathbf{n}_1). \quad (6)$$

Applying the standard Kalman filter framework (replace the time index t with scale index l), given Eq. 5 and Eq. 6, an optimal estimator for $\mathbf{v}(l)$ is derived from the estimate of the coarse scale $\hat{\mathbf{v}}(l-1)$ and a set of fine scale derivative measurements:

$$\begin{aligned}
\hat{\mathbf{v}}(l) &= \mathbf{E}(l-1)\hat{\mathbf{v}}(l-1) + K(l)\nu(l), \\
\mathbf{\Lambda}(l) &= \mathbf{\Lambda}'(l) - K(l)\nabla\mathbf{f}^T(l)\mathbf{\Lambda}'(l), \\
K(l) &= \mathbf{\Lambda}'(l)\nabla\mathbf{f}(l)\cdot \\
&\quad [\nabla\mathbf{f}^T(l)(\mathbf{\Lambda}'(l) + \mathbf{\Lambda}_1)\nabla\mathbf{f}(l) + \Lambda_2]^{-1}, \\
\nu(l) &= -f_t(l) - \nabla\mathbf{f}^T(l)\mathbf{E}(l-1)\hat{\mathbf{v}}(l), \\
\mathbf{\Lambda}'(l) &= \mathbf{E}(l-1)\mathbf{\Lambda}(l-1)\mathbf{E}(l-1)^T + \mathbf{\Lambda}_0. \quad (7)
\end{aligned}$$

The innovation $\nu(l)$ is approximated as the temporal derivative of the warped images. More detail about this approximation process can be found in [14].

4.1.2 Region-based Parametric Model Fitting

Eero Simoncelli's approach [15] tends to over-smooth the solution due to:

1. uniform window size for defining a neighborhood (a fixed weighting window size of 3×3 is used in [15]),
2. level to level propagation of information.

One solution to the problem is to use window sizes that are adaptive to the local image properties. Given that information propagation is actually the desirable property of a multi-scale approach, it is hard to address the over-smoothing problem caused by level to level information propagation. To solve this problem, we take inspiration from [2] by making use of parametric model to fit flow vectors to regions from image segmentation. It is commonly assumed that motion of the pixels within the same region can be fitted to a parametric model. For each pixel, denoted by a its coordinates, $\mathbf{x} = (x_i, y_i)$, within the same region, one of the following models is selected by the algorithm to fit flow vectors:

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_i) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
\mathbf{a} &= \begin{bmatrix} a_0 & a_3 \end{bmatrix}, \\
\mathbf{F}(\mathbf{x}_i) &= \begin{bmatrix} 1 & x_i & y_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_i & y_i \end{bmatrix}, \\
\mathbf{a} &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \end{bmatrix}, \\
\mathbf{F}(\mathbf{x}_i) &= \begin{bmatrix} 1 & x_i & y_i & x_i^2 & x_i y_i & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i y_i & y_i^2 & 1 & x_i & y_i \end{bmatrix}, \\
\mathbf{a} &= \begin{bmatrix} a_0 & a_1 & a_2 & a_6 & a_7 & a_3 & a_4 & a_5 \end{bmatrix}.
\end{aligned}$$

The two-parameter model corresponds to translation, six-parameter model corresponds to affine motion and the eight-parameter model corresponds to quadratic motion.

Minimizing the following weighted least squares equation gives the estimate of the model parameters $\mathbf{a}_r$ for region r,

$$\hat{\mathbf{a}}_r = \arg\min_{\mathbf{a}_r} \sum_i^r (\mathbf{v} - \mathbf{F}(\mathbf{x}_i)\mathbf{a}_r)^T \Lambda_{\mathbf{v}}^{-1} (\mathbf{v} - \mathbf{F}(\mathbf{x}_i)\mathbf{a}_r). \quad (8)$$

Though this formulation is similar in spirit to that of [2], the robust error norm is not used as we have an uncertainty model for $\mathbf{v}$ from Simoncelli's approach [14]. Pixels in the region with reliable flow $\mathbf{v}$ carry more weight in the fitting process. These pixels correspond to edge pixels or the regions with rich texture. Hence the fitting is more robust.

We use this simple weighted least square by combining region-fitting with Simoncelli's approach. The cost function of Eq. 8 is still convex and guaranteed to have an optimal solution given enough pixels in the region. Let $\hat{\mathbf{a}}$ be the optimal solution, the updated flow field $\hat{\mathbf{v}}'$ and corresponding covariance $\Lambda'_{\mathbf{v}}$ are computed as following:

$$\begin{aligned}
\hat{\mathbf{v}}' &= \mathbf{F}(\mathbf{x}_i)\hat{\mathbf{a}}, \\
\Lambda_{\mathbf{a}} &= (\mathbf{J}(\mathbf{x}_i)^T \Lambda_{\mathbf{v}}^{-1} \mathbf{J}(\mathbf{x}_i))^{-1}, \\
\Lambda'_{\mathbf{v}} &= \mathbf{F}(\mathbf{x}_i)\Lambda_{\mathbf{a}}\mathbf{F}(\mathbf{x}_i)^T, \quad (9)
\end{aligned}$$

where $\mathbf{J}(\mathbf{x}_i)$ is the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{x}_i$.

In the combined approach, first image segmentation based on color/intensity information is performed via mean

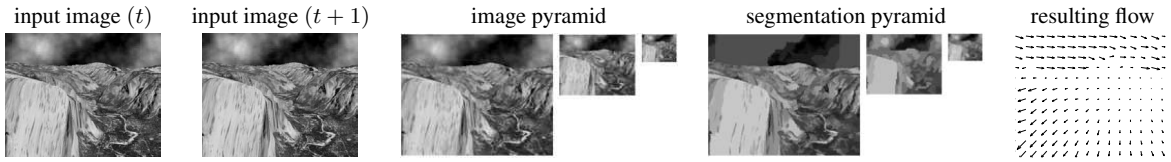

Figure 2. *Example of flow computation for Yosemite sequence [6].*

Figure 3. *Example of disparity computation for Teddy data set [18].*

shift [3] at each resolution level of the image pyramid. The order of the parametric model used for fitting is adaptive to the resolution level, region size and fitting residual error. Lower order model is preferred if an higher order model fails to improve the fitting quality. When the residual error of fitting a eight-parameter model is still high and the region size is large, the region is split by using mean shift [3] on the region flow field as color/intensity information alone is not enough. Model fitting is then performed on the newly split regions. This step can be recursive; the stopping criteria is either the region is small enough or the error residual is below a threshold. Figure 2 shows the process of computing optical flow for the Yosemite sequence [6].

4.2. Distribution of Disparity

The same algorithm for optical flow computation is applied for computing disparity of input image pair captured by the stereo rig. Just substitute the time index t and $t+1$ with the view index l and r, where l refers to *left* view and r refers to the *right* view in a binocular stereo rig. Only horizontal displacement and corresponding variances are computed. Most researchers treat optical flow and disparity computation differently as the constant brightness assumption is often violated in disparity computation. By using multi-scale based approach, the problem can be solved in the same way as optical flow. Figure 3 shows the process of computing disparity and disparity obtained for the Teddy data set [18].

4.3. Computing 3D scene flow

The Camera rig is fixed in our system, so there is no camera motion. Following [19], scene flow is defined as the 3D motion field of the points in the world, just as optical flow is the 2D motion field of the points in an image. Any optical flow is simply the projection of the scene flow onto the image plane of a camera.

Given a 3D point $\mathbf{X} = (X, Y, Z)$, the 2D image of this point in view v is denoted as $\mathbf{x}_v = (x, y)$. The 2D components of $\mathbf{x}_v$ are

$$x_v = \frac{[\mathbf{P}_v]_1(X,Y,Z,1)^T}{[\mathbf{P}_v]_3(X,Y,Z,1)^T}, \quad y_v = \frac{[\mathbf{P}_v]_2(X,Y,Z,1)^T}{[\mathbf{P}_v]_3(X,Y,Z,1)^T}, \tag{10}$$

where $[\mathbf{P}_v]_j$ is the j^{th} row of the projection matrix $\mathbf{P}_v$. If the camera is not moving, then $\mathbf{v} = \frac{d\mathbf{x}_v}{dt}$ is uniquely determined by the following:

$$\frac{d\mathbf{x}_v}{dt} = \frac{\partial \mathbf{x}_v}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dt}. \tag{11}$$

To solve for the scene flow $\mathbf{V} = \frac{d\mathbf{X}}{dt}$, two equations are needed. Hence at least two cameras are needed. The setup of the system of equations is simply

$$\mathbf{BV} = \mathbf{U}, \tag{12}$$

where

$$\mathbf{B} = \begin{bmatrix} \frac{\partial x_{v_1}}{\partial X} & \frac{\partial x_{v_1}}{\partial Y} & \frac{\partial x_{v_1}}{\partial Z} \\ \frac{\partial y_{v_1}}{\partial X} & \frac{\partial y_{v_1}}{\partial Y} & \frac{\partial y_{v_1}}{\partial Z} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial x_{v_N}}{\partial X} & \frac{\partial x_{v_N}}{\partial Y} & \frac{\partial x_{v_N}}{\partial Z} \\ \frac{\partial y_{v_N}}{\partial X} & \frac{\partial y_{v_N}}{\partial Y} & \frac{\partial y_{v_N}}{\partial Z} \end{bmatrix}, \mathbf{U} = \begin{bmatrix} \frac{\partial x_{v_1}}{\partial t} \\ \frac{\partial y_{v_1}}{\partial t} \\ \cdot \\ \cdot \\ \frac{\partial x_{v_N}}{\partial t} \\ \frac{\partial y_{v_N}}{\partial t} \end{bmatrix}. \tag{13}$$

A singular value decomposition of $\mathbf{B}$ gives the solution that minimizes the sum of least squares of the error obtained by re-projecting the scene flow onto each of the optical flows.

4.4. Integrated Approach

As discussed in Section 2, it is known that the correspondence problem (across different views or across different time frames) is ill-posed. Hence it is hard to compute scene flow reliably from optical flow. One way to get around this is to use many cameras, as reported in [19], a total number of 51 cameras were used to solve Eq. 12 reliably.

Instead of aiming to improve the accuracy by using more cameras, we propose to incorporate the covariances derived from the computation of optical flow and disparity. By taking the covariances from disparity and optical flow into account, the linear system of Eq. 12 tends to produce reasonable scene flow given a small number of cameras. The estimated scene flow with covariances can be used for applications like probabilistic 3D tracking and 3D motion and structure analysis.

For a stereo pair, the 3D coordinate $\mathbf{X}$ is related to the disparity d and corresponding image coordinates $\mathbf{x}_{v_l}$ and $\mathbf{x}_{v_r}$ where v_l indicates left view and v_r indicates right view. Let T denote the baseline and f denote the focal length (both cameras are assumed to have the same focal length). The following equation defines the relationship between the 3D coordinates, 2D image coordinates in the left and right cameras and the pixel disparity between left and right cameras.

$$X = \frac{T(x_{v_l} + x_{v_r})}{2d},\ Y = \frac{T(y_{v_l} + y_{v_r})}{2d},\ Z = \frac{fT}{d}. \quad (14)$$

Hence we solve Eq. 13 for scene flow, $\mathbf{V}$ by:

$$\hat{\mathbf{V}} = \arg\min_{\mathbf{V}} (\mathbf{BV} - \mathbf{U})^T \mathbf{W}^{-1} (\mathbf{BV} - \mathbf{U}), \quad (15)$$

where

$$\mathbf{W} = \Lambda_{\mathbf{d}} \Lambda_{\mathbf{v}}. \quad (16)$$

By covariance propagation, the covariance of $\mathbf{V}$ is:

$$\Lambda_{\mathbf{V}} = (\mathbf{B}^T \mathbf{W}^{-1} \mathbf{B})^{-1}. \quad (17)$$

Algorithm 1 Algorithm for computing 3D scene flow

initialize Λ_p and Λ_0 to small value.
for $l = 0$ to $L - 1$ **do**
  segment $f_{v_l}(t, l)$ and $f_{v_r}(t, l)$ via mean shift[3],
  if $l == 0$ **then**
    compute $\Lambda_{\mathbf{u}}(l)$, $\hat{\mathbf{u}}(l)$, $\Lambda_d(l)$ and $\hat{d}(l)$ [Eqs.3 and 4],
  else
    compute $\Lambda_{\mathbf{v}(l)}$, $\hat{\mathbf{v}}(l)$, $\Lambda_d(l)$ and $\hat{d}(l)$ [Eq.7],
  end if
  do model fitting as described in Section 4.1.2,
  compute $\hat{\mathbf{u}}'$, $\Lambda'_{\mathbf{u}}$, $\hat{d}'$ and Λ'_d [Eq. 9],
  set $\hat{\mathbf{u}}(l) = \hat{\mathbf{u}}'(l)$, $\Lambda_{\mathbf{u}} = \Lambda'_{\mathbf{u}}$, $\hat{d}(l) = \hat{d}'(l)$, $\Lambda_d(l) = \Lambda'_d(l)$,
  if $l == 0$ **then**
    solve $\hat{\mathbf{V}}(l)$ [Eq. 15],
  else
    solve $\hat{\mathbf{V}}(l)$ [Eq. 15], using $\hat{\mathbf{V}}(l-1)$ as the initial estimate,
  end if
end for

To compute the scene flow for two consecutive frames in the stereo video streams, we use f_{v_l} to denote the left video stream and f_{v_r} to denote the right video stream. First we build image pyramids of height L for $f_{v_l}(t)$, $f_{v_l}(t+1)$, $f_{v_r}(t)$ and $f_{v_r}(t+1)$. Pyramid images are indexed by l where $l = 0$ is the index for image at the lowest resolution level and $l = L - 1$ is the index for image at the highest resolution level. The optical flow fields computed at each level of the pyramid for the binocular views are denoted as $\mathbf{v}_{v_l}(l)$ and $\mathbf{v}_{v_r}(l)$. Disparity is denoted as $d(l)$. Algorithm 1 describes a single integrated method for computing optical flow, disparity and 3D scene flow.

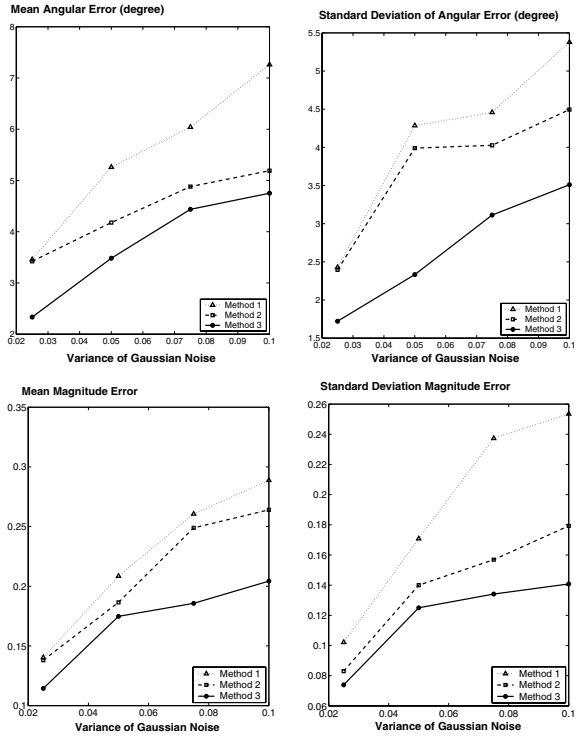

Figure 4. *Angular error (first row) and magnitude error (second row) of synthetic data with added Gaussian noise.*

5. Experiments

Two sets of experiments are conducted to demonstrate the effectiveness of the weighted least square model and the performance of the algorithm.

5.1. Synthetic 3D Data

To show the effectiveness of the weighted least squares method, 3600 3D points on a planar surface with known 3D scene flow, 2D optical flow and disparity are generated. The point-wise 3D scene flow is drawn from a Gaussian distribution. Each point moves in slightly different direction with different magnitude which corresponds to non-rigid motion. Gaussian noise with different variances are added to the 2D optical flow and disparity. Three methods are tested. Accuracy of the computed 3D scene flow is measured using the

average angular error and average magnitude between computed 3D scene flow and known 3D motion. The mean and standard deviation of the angular and magnitude error of the estimated 3D scene flow are reported based on the average of 10 runs of the experiments.

Method 1: Eq. 12 without incorporating covariance [19].

Method 2: Eq. 15 where only the covariance of 2D optical flow is used.

Method 3: Eq. 15 where both the covariance of 2D optical flow and the variance disparity are used.

Figure 4 shows the mean and standard deviation of angular and magnitude error. It is clear that by incorporating the covariance of the 2D optical flow and the variance of the disparity, more accurate 3D scene flow can be estimated via the weighted least squares.

5.2. Real Scene

To evaluate the algorithm in practice, experiments are performed on real scene sequences. The first row of Figure 5 shows frames from a binocular video sequence captured for the experiment. The sequences were captured with Videre MEGA-D system: a binocular stereo camera connected with Matrox capture card through fire wire cable. The frame rate of stereo sequence is around 30 frames/sec with resolution of 320×240. The scene flow algorithm is implemented Matlab and C++. Experiments were conducted on an AMD Athlon MP 2100+ machine. Dense scene flow is computed for each frame in about 2 minutes per frame. The sequences acquired are rectified and the calibration information is known.

The binocular video sequences are acquired in an uncontrolled illuminated environment, hence the estimates of optical flow and disparity are noisy. The observable motion in the scene is the backward movement of right hand and the forward movement of left hand. The second row of Figure 5 shows the 2D projection of the 3D flows in the left and right view, the Z velocities and the variances. From the result, we can see that the 3D movements of the left and right hands have been described reliably. The variance of the Z velocity gives information of how reliable is the estimate. Darker areas indicate lower variance and brighter areas represent higher variance. The variance is tied to the 2D image properties, e.g. local image contrast and texture information. We get comparable results to those of [24] in a similar setup, while they used three cameras and we only use a binocular camera rig.

The third and fourth rows of Figure 5 show results of another sequence where both hands move forward. The last row of Figure 5 shows the results without accounting for covariances (Eq. 12). It is clear that the Z velocities recovered are very noisy compared with the results obtained from our algorithm.

6. Discussion and Conclusions

A multi-scale integrated algorithm for 3D scene flow computation is proposed in this paper. Covariances and variances from the probabilistic framework for optical flow and disparity computation are combined to estimate 3D scene flow. Experiments with synthetic and real data demonstrate good performance with just two cameras. Another benefit of the framework is that we can get covariances of the estimated 3D scene flow. The covariances are derived from the 2D image data and give a measure of how reliable the estimated flow is. The covariances can provide a good initialization for model based tracking algorithms. Our future work includes: (1) incorporating the output from our framework in tracking applications such as vision-based human-computer interfaces; (2) analyzing and annotating events in video through analysis of 3D scene flow.

References

[1] S. Avidan and A. Shashua. Non-rigid parallax for 3D linear motion. In *CVPR*, pages 62 – 66, 1998.

[2] M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972 – 986, 1996.

[3] D. Comaniciu and P. Meer. Meanshift analysis and applications. In *Proc. ICCV*, 1999.

[4] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159 – 179, 1998.

[5] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *CVPR*, 2003.

[6] D. J. Heeger. Model for the extraction of image flow. *J. Opt. Soc. Am. A*, 4(8):1455–1471, 1987.

[7] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981.

[8] W.-H. Liao, S. J. Aggrawal, and J. K. Aggrawal. The reconstruction of dynamic 3D structure of biological objects using stereo microscope images. *Machine Vision and Applications*, 9:166 – 178, 1997.

[9] B. D. Lucas and T. Kanade. An interative image registration technique with an application to stereo vision. In *IJCAI*, pages 674 – 679, 1981.

[10] S. Malassiotis and M. G. Strintzis. Model-based joint motion and structure estimation from stereo images. *CVIU*, 65(1):79 – 94, 1997.

[11] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*, 15(6):580 – 591, 1993.

[12] A. P. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *PAMI*, 13(7):730 – 742, 1991.

[13] Y. Q. Shi, C. Q. Shu, and J. N. Pan. Unified optical flow field approach to motion analysis from a sequence of stereo images. *Pattern Recognition*, 27(12):1577 – 1590, 1994.

[14] E. P. Simoncelli. Bayesian multi-scale differential optical flow. In B. Jahne, H. Haussecker, and P. Geissler, editors, *Handbook of Computer Vision and Applications*, chapter 14, pages 397 – 422. Academic Press, 1999.

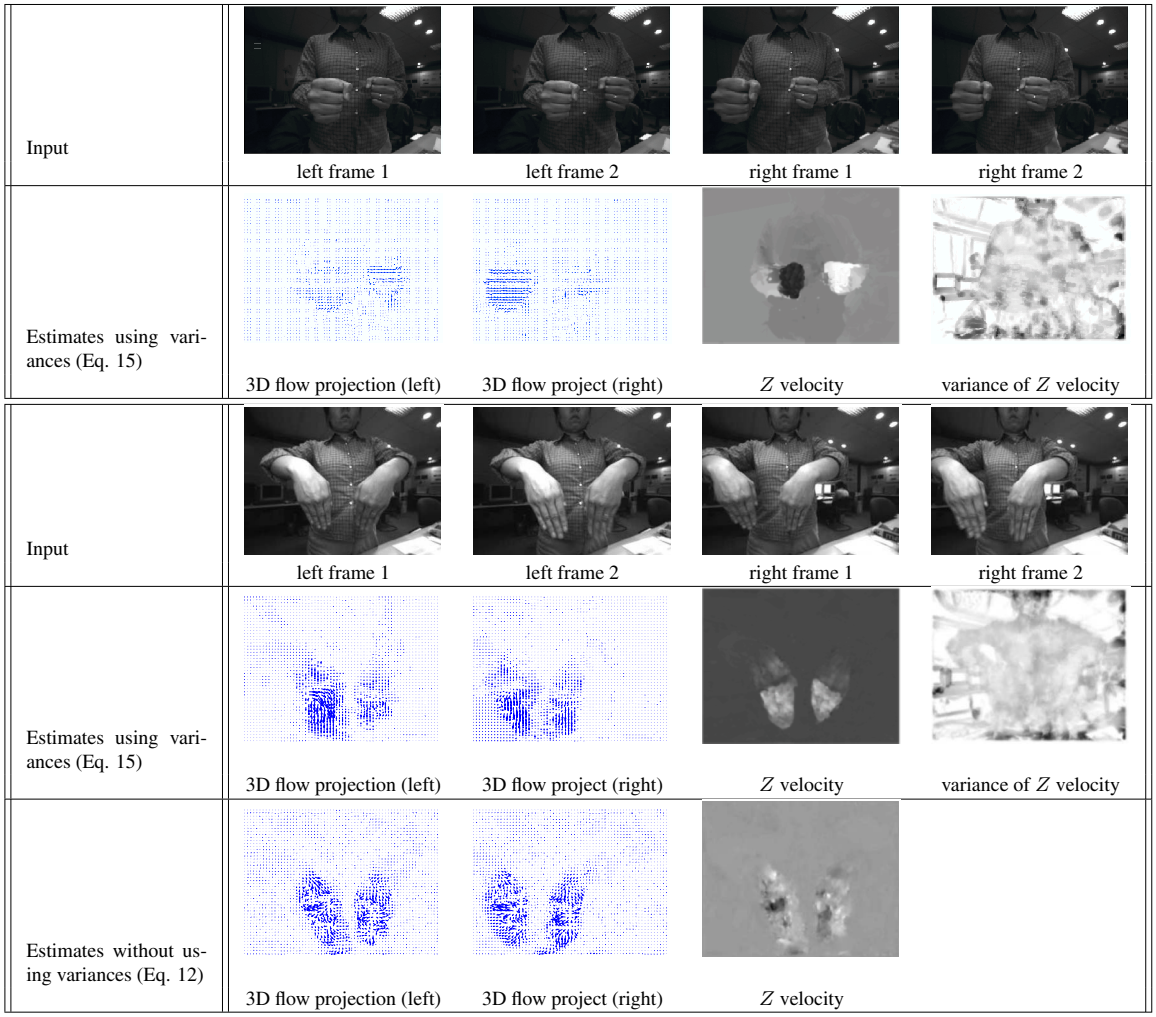

Figure 5. *Experimental results with real scene. In the Z velocity intensity image, the darker area represents the hand moving away from the camera, the brighter area indicates the hand moving towards the camera. The last row show the results for the second sequence using Eq.12.*

[15] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *CVPR*, 1991.

[16] S. Ullman. *The interpretation of Visual Motion*. MIT Press, 1979.

[17] S. Ullman. Maximizing the rigidity: The incremental recovery of 3-D shape and nonrigid motion. *Perception*, 13:730 – 742, 1984.

[18] http://www.middlebury.edu/stereo.

[19] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. ICCV*, volume 2, pages 722 – 729, 1999.

[20] A. M. Waxman and J. H. Duncan. Binocular image flows: Steps toward stereo-motion fusion. *PAMI*, 8(6):715 – 729, 1986.

[21] G. S. Young and R. Chellappa. 3-D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness. *PAMI*, 12(8):735 – 759, 1999.

[22] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *CVPR*, 2003.

[23] Y. Zhang and C. Kambhamettu. Integrated 3D scene flow and structure recovery from multiview image sequences. In *CVPR*, 2000.

[24] Y. Zhang and C. Kambhamettu. On 3D scene flow and structure estimation. In *CVPR*, 2001.

[25] Z. Zhang and O. Faugeras. Estimation of displacements from two 3-D frames obtained from stereo. *PAMI*, 14(12):1141 – 1156, 1992.