Fusing Multiple Two Frame Depth Estimates for 3D Reconstruction with Unknown Noise Distribution

Amit K. Roy Chowdhury
R. Chellappa
Center for Automation Research
and
Department of Electrical and Computer Engineering
University of Maryland, College Park
MD 20742, USA
{amitrc,chella}@cfar.umd.edu

Abstract

The problem of structure from motion (SFM) is to extract the three-dimensional model of a moving scene from a sequence of images. Traditional SFM algorithms use just two images but produce inaccurate reconstructions of the 3D scene because of incorrect estimation of the motion. Recently, algorithms have been proposed that use multiple (> 2) frames. This paper proposes a computationally efficient framework for estimating structure from an image sequence taking into consideration the error in the 2-frame estimates, but without recourse to strong statistical assumptions on the observations. We propose a simple way to model the observations, which are the depth estimates obtained from traditional 2-frame SFM algorithms, and present a batch and a recursive solution to the multiframe estimation process. Our algorithms use bootstrapping and stochastic approximation methods. We also propose a method to estimate the number of frames to be used in the recursive solution using the Fisher information criterion. The algorithms were primarily applied to model human face and a number of experimental results are reported.

Key Words: structure from motion (SFM), multi-frame SFM, modeling depth observations, bootstrapping, Robbins-Monro stochastic approximation, 3D face modeling.

1 Introduction

The problem of structure from motion (SFM) is to extract the three-dimensional model of a moving scene from a sequence of images. Traditional SFM algorithms [1], [7], [16] recover a 3D scene structure from two images. However, these algorithms often produce inaccurate reconstructions of the scene, mainly due to incorrect estimation of camera motion. Dutta and Snyder [5] show that even small errors in estimating the camera motion parameters can cause large errors in the 3D reconstruction.

Recently techniques have been developed that use multiple images for scene reconstruction, achieving greater robustness and accuracy by using additional information. The techniques for multi-frame structure from motion (MFSFM) can be classified into two groups: batch methods in which all the images are processed simultaneously, and incremental methods, in which the images are processed sequentially and the reconstruction estimates are improved iteratively with every image processed. Thomas and Oliensis [24] describe an incremental algorithm that estimates the structure and corrects for the error in estimating the camera motion. Szeliski [21] developed an algorithm using energy function minimization where the observed and true depth values were related by a linear, statistical model.

This paper describes two new algorithms for MFSFM obtained by fusing the two frame estimates and reports the experimental results on a large number of image sequences, mostly used in 3D modeling of human faces. However, the method is general enough to be used on other types of image sequences as well as in other applications with very little modification. The correspondence problem is not addressed in this paper. We assume that the 2-frame estimates are available from a suitable 2-frame SFM algorithm [1] and present a batch (using bootstrapping [6]) and recursive (using stochastic approximation [18]) framework for fusing these estimates into a single one. The observed depth values obtained from two frame SFM algorithms are modeled as the true value embedded in noise. We show that it is possible to reconstruct the depth to arbitrary accuracy without taking recourse to the strong assumptions of Gaussianity and independence of the observations which cannot be validated for a large number of cases. We also show that it is possible to estimate the number of frames necessary for fusion by calculating the Fisher information criterion at every step.

2 Review of MFSFM

As has been mentioned earlier, there are two broad classes of MFSFM algorithms, batch and recursive methods. Typical batch methods formulate the problem of estimating the structure as one of minimizing an objective function defined as a sum of squares of the differences between the actual observed images and the projections of their estimated 3D locations, over all tracked positions and images [4], [8], [9], [21], [22], [23]. However, the objective function is nonlinear and avoiding local minima becomes a very difficult task. Incremental methods update the 3D model as new images are acquired. Though computationally feasible and practically implementable, incremental algorithms require a reliable estimate of the error in the model and can perform very poorly if the error is not modeled accurately. Thomas and Oliensis [24] describe a method of estimating the error, which is modeled as a combination of the error in the estimated camera motion and the error in tracking the image coordinates, assuming the image noise to be independent and zero mean.

In our model of the two-frame depth estimates, which we will henceforth refer to as the observations, we follow Szeliski's idea [21] that the true values are embedded in additive noise but show that it is possible to solve the problem without the assumptions of Gaussianity or independence of the observations. Also, our objective function takes into account the fact that some of the observations may be highly erroneous and need to be ignored. We also show that it is possible to estimate the number of images that need to be considered for a reliable result based on the estimation of the Fisher information criterion. Finally, our estimates are able to reconstruct the 3D model to an arbitrary level of accuracy given a sufficient number of images.

[1] The particular two-frame algorithm chosen here was the one described in [16] because of its speed of computation, but our fusion algorithm is not bound by this particular method.

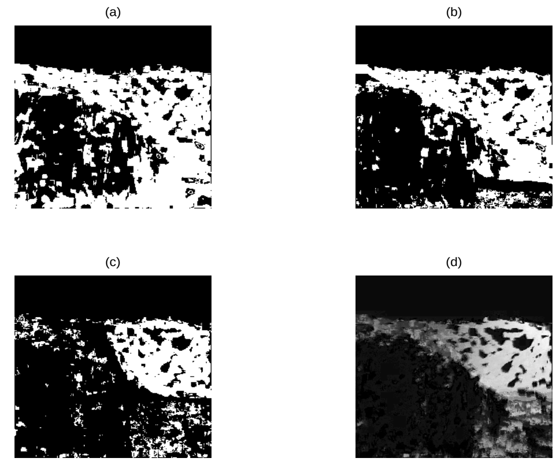

Figure 1: **(a),(b) and (c) represent three two-frame depth maps chosen randomly from the Yosemite sequence; (d) represents the fused depth map from 15 frames. The fused estimate is much closer to the true value.**

Since our algorithm fuses two-frame reconstructions, its robustness is limited by the robustness of these reconstructions [12]. Except for cases where there is very small camera translation or few tracked points, [12] argues that fusing two-frame estimates is a robust and accurate method. To motivate the point that data fusion produces considerably better results, we consider an example on the synthetic Yosemite sequence. Fig. 1 shows the depth maps for three two-frame estimates and for the fused estimate from 15 frames. The fused estimates are closer to the true value.

3 Overview of the Algorithm

In this section, we outline the observation model and the cost function sought to be minimized in order to obtain the estimate.

3.1 The Observation Model

Let $\mathbf{d_i}$ represent the depth value obtained by the two-frame SFM algorithm from the i and $(i+1)$-th frame. Let the n-th position in that vector be denoted by $\mathbf{d_i}(n)$, representing the depth at the n-th feature point (for a feature point based algorithm) or the n-th pixel for a flow based algorithm, with a lexicographic ordering of the image being considered for ease of notation (our implementation corresponds to the latter case). We will henceforth denote the sequence of vector observations of the two-frame depth estimates by $\{\mathbf{d_i}, i = 1, ..., K\}$, $(K+1)$ being the total number of frames. For ease of description, we will use the term depth at a pixel and depth at a (feature) point interchangably.

Our observation model assumes that the two-frame depth measurements are related to the actual depth value $\mathbf{u}$ by the following linear transformation

$$\mathbf{d_i} = \mathbf{H}\mathbf{u} + \nu_{\mathbf{i}}, \qquad i = 1, ..., K \tag{1}$$

where $\nu_{\mathbf{i}}$ is a noise process associated with the observation vector. At this moment we do not assume any distribution for the noise nor independence of the noise vector $\nu_{\mathbf{i}}$ across i. The model is very similar to the one assumed in previous multi-frame SFM fusion work [9], [21]. However, in both these cases, Gaussianity and statistical independence across time were assumed. $\mathbf{H}$ is assumed to be a constant sparse matrix, implying that the depth at a point relies on only a few neighbors. A particular simplifying assumption that can be used is a diagonal structure for $\mathbf{H}$ [9]. This basically implies that the depth at every pixel is treated independently. For the feature point based method, we cannot do much better as depth at only a sparse set of points is available; for the flow based method, the difference between choosing a very sparse but non-diagonal $\mathbf{H}$ and a diagonal structure is not much in the final result, because we can model dependencies at only neighboring pixels and motion estimates are usually poor over a region [17]. A more complicated structure on $\mathbf{H}$ is not practical because of the increase in computational burden and the inability to decide on a suitable structure *a priori*. Because of this form of $\mathbf{H}$, we will henceforth consider the depth at each pixel separately. Thus, we can write, for every n,

$$d_i = u + \nu_i, \qquad i = 1, ..., K \tag{2}$$

(Note that we have dropped the bold script indicating vector notation.) [21] assumes that u and ν_i are Gaussian random variables, with suitable mean and covariances. Under this assumption, given $u = u$, the observation vector $\mathbf{d} = [d_1, ..., d_K]$ is also Gaussian. For Gaussian random variables, all odd central moments are identically zero (actually true for any symmetric distribution) and all cumulants of order greater than two are zero [13]. Fig. 2 shows the plot of the estimates of the central moments and cumulants of the observation vector. Analysis of the plots reveal that the properties of Gaussianity of the observations are not satisfied. These plots are for a particular sequence but similar results are obtained for other sequences also. Hence, we feel that the assumption of Gaussianity is difficult to justify in many cases.

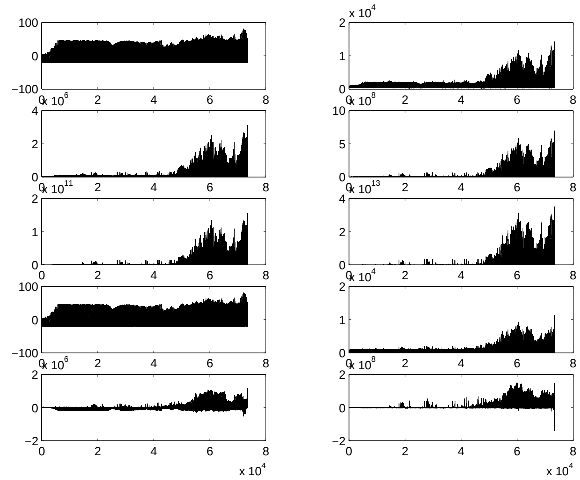

Figure 2: **The top six figures plot the estimates of the first six moments of the observation vector and the bottom four figures plot the first four cumulants. The horizontal axis represents the pixel number. The first column represents the odd central moments/cumulants and the second column the even ones. It is apparent that the assumption of Gaussianity of the observation vector is questionable for a large number of pixels.**

3.2 The Cost Function

The form of our observation model corresponds to the classical linear regression model, whereby we try to minimize the mean square error

$$MSE(u) = \sum_{i=1}^{K} (d_i - u)^2 \tag{3}$$

which yields the solution that the optimal u, denoted by $u^* = \sum_{i=1}^{K}(d_i)/K$ [15]. However, sample means are sensitive to influential values. An analysis of the depth values across frames for any pixel shows that there are some values which can be characterized as outliers and should not be considered during fusing the estimates. Fig. 3 shows a plot of the depth values across 50 frames for four randomly chosen points. It can be seen that there are isolated outliers in all the four cases.

It is a well-known fact that the median is less sensitive to outlying data points than the mean. More formally, given n data points $\mathbf{x} = (x_1, ..., x_n)$, the breakdown of an estimator $s(\mathbf{x})$ is said to be m/n where m is the smallest number such that if we are allowed to

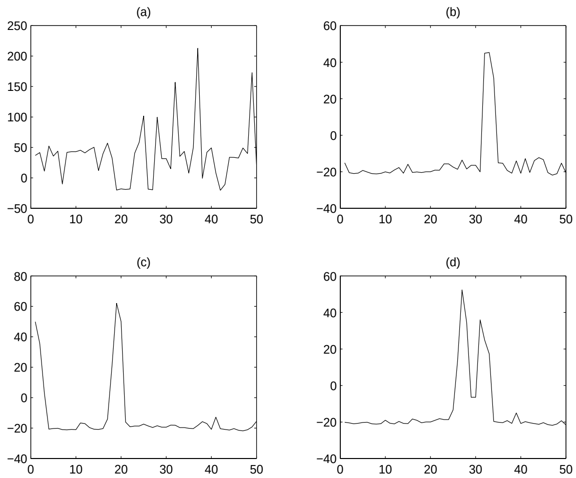

Figure 3: **A plot of the depth values across 50 frames for four randomly chosen points. It can be seen that there are isolated outliers in all the four cases.**

change m data values in any way, we can force the absolute value of s for the "perturbed" sample towards plus or minus infinity. High breakdown is good, with 50% being the largest value that makes sense (a larger value does not make sense as it is not clear which are the good points and which are the bad). The mean of a sample has breakdown $1/n$ as by changing just one data value we can force the sample mean to have any value whatsoever. The sample median has breakdown 50%, reflecting the fact that it is less sensitive to individual values. The least squares regression estimator inherits the sensitivity of the mean and has breakdown $1/n$, whereas the least median of squares estimator (LMS) (defined as the median of the squares of the error) has breakdown roughly 50% [6]. For our problem, we use the LMS criterion.

$$u^* = \min_u \left(median(d_i - u)^2 \right) \tag{4}$$

The disadvantage of this method is that we no longer have a neat formula as we had for the mean-square error criterion. Sections 4 and 5 describe the method of solving for this estimate.

4 A Batch Algorithm Using Bootstrapping

In this section we explain our method of solving the estimation problem outlined in the previous section using the bootstrapping technique.

Bootstrapping techniques, introduced by Efron [6], are non-parametric estimation methods for the statistical behavior of an estimate $\hat{\theta} = s(\mathbf{x})$. The data points $\mathbf{x} = (x_1, ..., x_n)$ are assumed i.i.d. from an unknown distribution F. The fundamental idea of bootstrap is to replace F by $\hat{F}$, the empirical distribution of the data. The basic steps in the algorithm are:

- Construct the empirical distribution $\hat{F}$ of F from the given samples.
- Draw independently B bootstrap samples $\mathbf{x}^{*\mathbf{1}}, ..., \mathbf{x}^{*\mathbf{B}}$ from $\hat{F}$ by means of random sampling with replacement.
- For each bootstrapped data set, compute the corresponding bootstrap estimate $\hat{\theta}^{*b} = s(\mathbf{x}^{*\mathbf{b}})$.

The set of bootstrap data yields a bootstrap distribution of $\hat{\theta}^*$ from which the statistical behavior of the estimate $\hat{\theta}$ can be inferred. The bootstrap estimated standard error of $\hat{\theta}$ can be computed as

$$\hat{e}_B(\hat{\theta}) = \left[\frac{\sum_{b=1}^{B} (\hat{\theta}^{*b} - \overline{\theta}^*)^2}{B-1} \right]^{1/2} \tag{5}$$

where $\overline{\theta}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^{*b}$. For further details, refer to [6].

Bootstrapping techniques have become popular in a number of areas, including computer vision [10]. The motivation for using this technique in our application of fusing the depth estimates is that in the case where real data is not normally distributed, bootstrap can improve on the classical normal approximation. Also, the bootstrap estimate obtained by the above algorithm will tend , as $B \to \infty$, to the plug-in estimate which is computed using the empirical distribution $\hat{F}$ instead of the unknown distribution F [6].

The number of bootstrap samples dictates the accuracy of the computation. As has been pointed out in [10] $B = 200$ and more than 20 measurements suffices to produce a good estimate. Efron [6] notes that $B = 50$ is often good enough and seldom do we need $B > 200$. For our problem, bootstrap data sets, $\mathbf{d}^{*b} = (d_1^{*b}, ..., d_K^{*b})$, were created as described before. The bootstrap replication $\hat{d}^{*b}$ were obtained as the minimizer of the median squared residual for the bootstrap data, i.e. $\min_u median(d_i^{*b} - u)^2$.

5 A Recursive Algorithm Using Stochastic Approximation

In this section, we present another method of obtaining the multi-frame depth estimates using Robbins-Monro (RM) stochastic approximation (SA) technique [11]. The algorithm is recursive (thus more practicable), does not assume any distribution of the observed data or their independence, and the obtained estimate is asymptotically unbiased and normal. Also, we describe a method of determining the number of frames necessary for obtaining a reasonable 3D reconstruction.

5.1 The Robbins-Monro Algorithm

The Robbins-Monro stochastic approximation (RMSA) algorithm is a stochastic search technique for finding the root θ^* to $g(\theta) = 0$ based on noisy measurements of $g(\theta)$. Specifically, let the measurements be $Y_k(\theta) = g(\theta) + e_k(\theta), k = 1, ..., K$, where $e_k(\theta)$ is assumed to be the noise term and K is the number of observations. The RMSA algorithm obtains the estimate by the following recursion,

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k Y_k(\hat{\theta}_k). \tag{6}$$

where a_k is an appropriately chosen sequence. Details of the algorithm can be found in [11], [18]. We will outline the method for obtaining the solution for our specific problem. Suppose that $F_X(x)$ is the unknown distribution of a sequence of observation $X_0, X_1, ...$ and we are interested in finding the root of the equation $g(\theta) = F_X(\theta) - 0.5 = 0$, i.e. the median of the distribution. For this problem, the RM recursion is as follows [18]:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k(s_k(\hat{\theta}_k) - 0.5) \tag{7}$$

where

$$s_k(\hat{\theta}_k) = \begin{cases} 1 & \text{if } X_k \leq \hat{\theta}_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The choice of the gain sequence a_k is determined by the convergence properties of the algorithm. We will not discuss the details here which can be found in [2], [18]. The commonly used gain sequence which has been found to be effective is $a_k = 0.1/(k+1)^{.501}$ and this was used in our experiments also.

In our implementation, a search set $\mathcal{U}$ of u was predetermined and the median was computed for each element of $\mathcal{U}$. The recursion was stopped after a suitable number of frames (a method for determining the number of frames will be described in the next section) and the minimum over $\mathcal{U}$, of the median of the square of the errors was computed, thus yielding u^*.

Stochastic approximation has a rich convergence theory [18]. It has been shown that the estimate $\hat{\theta}_k$ obtained by the RMSA procedure described above converges to the actual root (in this case the median) almost surely, under suitable conditions. This property of the estimate allow us to claim that given a sufficient number of frames, we can reconstruct the 3D model of the scene up to an arbitrary accuracy.

5.2 Estimating the Fisher Information

In any estimation problem it is necessary to determine the information provided by the measurements. One particular way of doing this is to look at the increase in the Fisher information [3]. Given the observations denoted by $\mathbf{Y}$, the Fisher information matrix is

$$J(\theta) = E_\theta[(\bigtriangledown_\theta \ln(f_\theta(\mathbf{Y})))(\bigtriangledown_\theta \ln(f_\theta(\mathbf{Y})))^T] \tag{9}$$

where θ is the parameter to be estimated given the observations, E_θ represents expectation with respect to θ and $\bigtriangledown_\theta$ represents the gradient with respect to θ.

Spall [19] describes a simple method for estimating the Fisher information using simultaneous perturbation for the gradient approximation and averaging for the expectation operation. We adopt this method with some modifications, as Spall uses the definition of the Fisher information involving the Hessians rather than the gradients. For the observation model $Y = \theta + X, X \sim f_X(x)$, where X is a random variable with a density f_X, (smaller case letters will denote realizations of the random variables denoted by the corresponding uppercase letters.), we can write

$$\begin{aligned} \frac{d}{d\theta} \log f_Y(y) &= \frac{d}{d\theta} \log f_X(y - \theta) \\ &= \frac{d}{dt} \log f_X(t) \frac{dt}{d\theta}, \qquad t = y - \theta \\ &= -\frac{1}{f_X(t)} \frac{df_X(t)}{dt}. \end{aligned}$$

To compute the gradient, we use the simultaneous perturbation form of gradient approximation [18]. Given an observation $y(\theta)$, the estimate of the gradient with respect to θ, $\hat{g}(\theta)$, is:

$$\hat{g}(\theta) = \frac{y(\theta + \Delta) - y(\theta - \Delta)}{2} \begin{bmatrix} \Delta_1^{-1} \\ \vdots \\ \Delta_p^{-1} \end{bmatrix} \tag{10}$$

where p is the dimension of θ, $\Delta = (\Delta_1, ..., \Delta_p)$ and the components of Δ are independent Bernoulli random variables (for details of this choice , refer to [18]). The steps in computing the Fisher information are:

Step 1 Given $\hat{\theta}_k$, generate a set of k pseudo measurements according to the empirical distribution of the observations. Denote these by $x_{pseudo}(k)$. Calculate the gradient according to (10). It may be necessary to average several gradient estimates with independent values of Δ. Compute the term within the expectation operator in the definition of Fisher information (9).

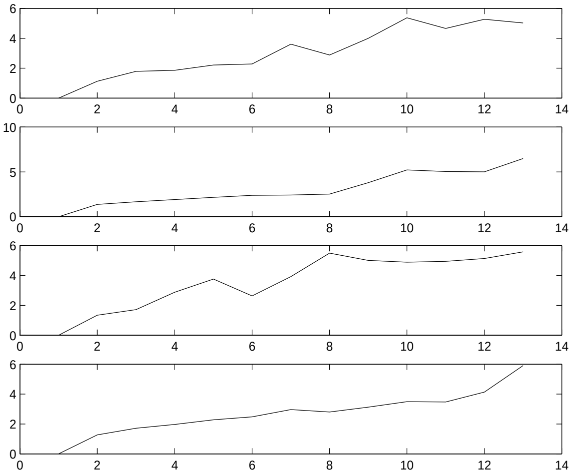

Figure 4: **The figure shows the variation of the Fisher information (FI) over increasing frames. The first few frames are neglected as we do not have enough information to compute the FI yet. The plots are for a random choice of four points, but the trends are very similar for the other points as well as for other image sequences.**

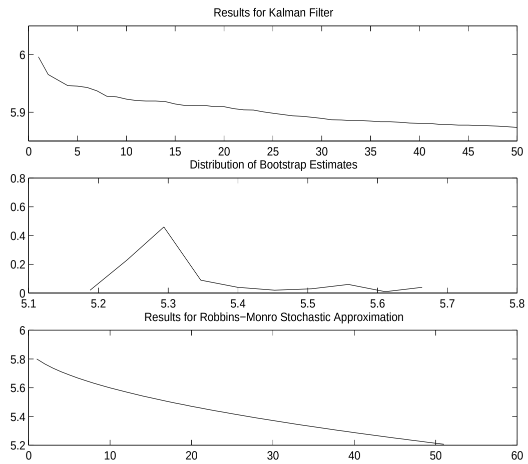

Figure 5: **(a): The estimates produced by the Kalman filter in succesive iterations; (b): The distribution of the bootstrap estimates; (c): The estimates produced by the RMSA algorithm in successive iterations.**

Step 2 Repeat Step 1 a large number of times, say N. Average the estimates obtained. This is the estimate of the Fisher information, $\hat{F}_k(\hat{\theta}_k)$.

To understand how we use the Fisher information to compute the number of frames over which to run the recursion, let us consider an example. Fig. 4 plots the variation of the Fisher information (FI) with increasing frames. The first few frames are neglected as we do not have enough information to compute the FI yet. The plots are for a random choice of four points, but the trends are very similar for the other points as well as for other image sequences. From these plots, we see that if we can set appropriate thresholds, we can determine when to stop the computation of the estimates (implying that the observations are not giving too much additional information). In our implementation, we followed a naive procedure of checking the differences between subsequent frames and if the difference is small for a few consecutive frames, we stop the estimation technique. However, there may be much better ways to exploit this information. As the computation is recursive, it fits in very well with the recursive RMSA framework.

6 Results and Analysis

In this section, we analyze the results of our algorithm. Most of our results are on human face image sequences. We apply the algorithms to create 3D face models. We also present a simulation result to highlight the advantage of using our algorithms compared to Kalman filtering when the noise is non-Gaussian.

6.1 Simulation Results

The purpose of our simulation is to prove that in the presence of non-Gaussian noise, our algorithms using bootstrapping and RMSA perform better than the traditional Kalman filter. This is to be expected as the Kalman filter is the optimal estimator in the presence of Gaussian noise only.

We consider a pixel whose true depth value is 5. The observations are corrupted by noise having a uniform distribution between $[0, 1]$. We consider 50 such observations. Fig. 5(a) plots the estimates produced by the Kalman filter at every step in the recursion. The estimates saturate at a value of 5.8. The distribution of the bootstrap estimates, plotted in 5(b), maximizes near 5.3. We used 50 bootstrap data sets. For the Robbins-Monro stochastic approximation, we plot the estimates at every stage of the recursion and the value obtained after 50 iterations is 5.2. Thus we see that for non-Gaussian noise our algorithms perform better than the Kalman filter. Similar results were obtained for other noise distributions also.

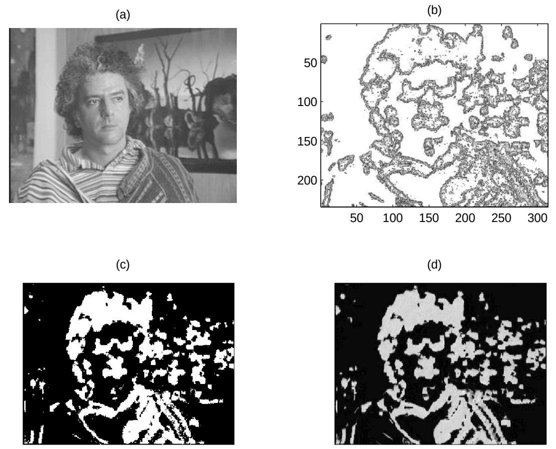

Figure 6: **(a): The first image in the sequence from which depth values are computed; (b): The level curves of the depth map; (c): The depth map using the bootstrapping method; (d): The depth map using the Robbins-Monro stochastic approximation method**

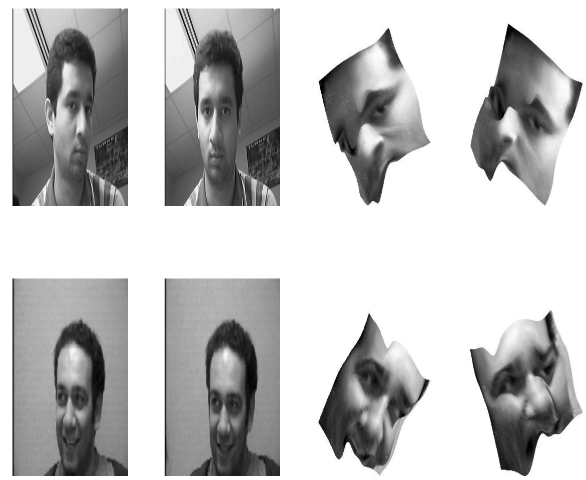

Figure 7: **Results of 3D modeling of human faces. The first and second columns show the first and last frames of the image sequence used to compute the depth map. The third and fourth columns depict the 3D models from camera positions not part of the original sequence.**

6.2 Results on Real Data

The first experiment with real data was conducted on an image sequence of Prof. Yiannis Aloimonos of the Computer Science Dept. A total of 30 frames was considered for this problem. The correspondence problem was not addressed in this work. The first plot in the Fig. 6 is of the first frame in the sequence. For the batch method, we used a total of 25 frames to obtain the multi-frame estimate. The depth values were constructed using both the methods described above. For the batch algorithm, a total of 50 bootstrap trials was considered. Fig. 6 shows the results obtained by this method. For the Robbins-Monro algorithm, using the Fisher information criterion described above, the number of frames varied between 15-25. The depth maps in both the cases seem to be satisfactory. Fig. 6(b) plots the level curves of the depth and the depth discontinuities can be judged from this plot. The result on the face shows that the discontinuities correspond to the expected ones for 3D face model.

Our next set of experiments were also aimed at 3D modeling of faces from an image sequence where the human subjects were asked to move their heads slightly. Fig. 7 plots the results obtained with the RMSA method. We emphasize more on the recursive strategy because of its practicability to process data as it is generated and also because of the theoretical reason that the assumptions on the data set are very mild and thus model the true situation better. The first and second columns show the first and last frames of the image sequence used to compute the depth map. The third and fourth columns show the 3D model for camera positions not part of the original sequence. [2]

7 Conclusion and Future Work

In this paper we have presented a batch and a recursive algorithm for fusing the two-frame depth estimates over multiple frames without taking recourse to the assumptions of Gaussianity and statistical independence. This we consider to be the main contribution of the paper following previous work in [21], [9]. We have also shown how the number of frames to be used can be computed from the Fisher information criterion. The work was applied to the modeling of human faces and the results have been presented.

References

[1] G.Adiv, Generating three-dimensional motion and structure from optic flow generated by several moving objects, *IEEE Trans. on PAMI*, 7(4), 1985, pp. 384-401

[2] A. Benveniste, M.Metivier and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, 1987.

[2] We have a working, real-time demo for modeling a human face from an image sequence in our lab.

[3] J.A.O'Sullivan, R.E. Blahut and D.L. Snyder, Information Theoretic Image Formation, *IEEE Trans. on Information Theory*, 44(6), October 1998.

[4] T.J. Broida and R. Chellappa, Estimating the kinematics and structure of a rigid object from a sequence of monocular images, *IEEE Trans. PAMI*, 13(6), 1991, pp. 497-513.

[5] R. Dutta and M. Synder, Robustness of correspondence-based structure from motion, *Proc. of 3rd Intl. Conf. on Computer Vision, Osaka, Japan, Dec. 1990*, pp. 106-110.

[6] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, 1993.

[7] A.R. Bruss and B.K.P. Horn, Passive Navigation, *Computer Vision Graphics Image Proc.*, 21(1), 1983, pp. 3-20

[8] R.V.Kumar, A.Tirmalai and R.C. Jain, A nonlinear optimization algorithm for the estimation of structure and motion parameters, *CVPR, San Diego, California, 1989* pp. 136-143

[9] L. Matthies, T. Kanade and R. Szeliski, Kalman Filtering algorithms for estimating depth from image sequences, *IJCV*, Vol.3, 1989, pp. 209-236.

[10] K. Cho, P. Meer and J. Cabrera, Performance Assessment through Bootstrap, TR CE-123, Rutgers University, Piscataway, NJ, 1995.

[11] H. Robbins and S. Monro, A Stochastic Approximation Method, Annals of Mathematical Statistics, vol 22, 1951, pp. 400-407.

[12] J. Oliensis, A Critique of Structure from Motion Algorithms, *NECI TR*, 1997.

[13] Papoulis, *Probability, Random Variables and Stochastic Processes*, Mc-Graw Hill, 1991.

[14] J. Oliensis, A Multiframe Structure from Motion Algorithm under Perspective Projection, *NECI TR*, 1997.

[15] V.Poor, Estimation and Detection Theory.

[16] S. Shridhar, Extracting Structure from Optical Flow Using Fast Error Search Technique, *CfAR Technical Report, University of Maryland*, CAR-TR-893, 1998.

[17] S. Shridhar and R. Chellappa, Robust Modeling and Estimation of Optical Flow with Overlapped Basis Functions, *CfAR Technical Report, University of Maryland*, CAR-TR-845, 1996.

[18] J.C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley.

[19] J.C. Spall, Resampling-based Calculation of the Information matrix for General Identification Problems, *Proc. of the American Control Conf., PA*, June 1998.

[20] P. Sturm and B. Triggs, A Factorization based algorithm for multi-image projective structure and motion, it ECCV, 1996, pp. 709-720.

[21] R. Szeliski, Estimating motion from sparse range data without correspondence, *2nd International Conference on Computer Vision*, 1988, pp 207-216.

[22] R.Szeliski and S.B.Kang, Recovering 3D shape and motion from image streams using non-linear least squares, *J. Visual Computation Image Representation.*, 5(1), 1994, pp. 10-28.

[23] C.J. Taylor, D. Kreigman and P. Anandan, Structure and motion in two dimensions from multiple images: A least squares approach, *Proc. of IEEE Workshop on Visual Motion, Princeton, NJ, 1991*, pp. 242-248.

[24] J.Inigo Thomas and J. Oliensis, Dealing with Noise in Multiframe Structure from Motion, *Computer Vision and Image Understanding*, Vol. 76(2), Nov. 1999, pp. 109-124.