COMPRESSED SENSING FOR MULTI-VIEW TRACKING AND 3-D VOXEL RECONSTRUCTION

Dikpal Reddy[†]*, Aswin C. Sankaranarayanan*[†]*, Volkan Cevher*[‡]*, Rama Chellappa*[†]

[†] Department of Electrical and Computer Engineering, University of Maryland, College Park

[‡] Department of Electrical and Computer Engineering, Rice University

ABSTRACT

Compressed sensing(CS) suggests that a signal, sparse in some basis, can be recovered from a small number of random projections. In this paper, we apply the CS theory on sparse background-subtracted silhouettes and show the usefulness of such an approach in various multi-view estimation problems. The sparsity of the silhouette images corresponds to sparsity of object parameters (location, volume etc.) in the scene. We use random projections (compressed measurements) of the silhouette images for directly recovering object parameters in the scene coordinates. To keep the computational requirements of this recovery procedure reasonable, we tessellate the scene into a bunch of non-overlapping lines and perform estimation on each of these lines. Our method is scalable in the number of cameras and utilizes very few measurements for transmission among cameras. We illustrate the usefulness of our approach for multi-view tracking and 3-D voxel reconstruction problems.

Index Terms— Compressed Sensing, Tracking, 3-D Voxel Reconstruction

1. INTRODUCTION

Visual camera networks are becoming popular with the increasing presence of cameras for surveillance, medical and smart room applications. In such a setting, it is important to design distributed algorithms that scale with the number of cameras and demand low communication overheads. By design, video cameras capture large amounts of data that is rich in structure and highly redundant. Further, in the context of smart cameras that have local processing capability it is possible to pre-process the acquired imagery before transmitting it over the communication channel. This allows us to extract the specific information from the videos over certain interesting epochs. The inherent structure in the acquired imagery (or the processed outputs) can be exploited to transmit only small number of measurements in order to address communication requirements.

In this paper, we utilize the emerging theory of compressed sensing(CS) [1, 2] to utilize the sparsity in our observations to represent and transmit the background subtracted images using small number of compressed measurements. The linear dependence of the desired parameters (location,visual hull) on the compressed measurements allow for direct reconstruction using ℓ_1 minimization based recovery algorithms [3, 4].

In prior work on multi-camera localization, silhouettes (foreground likelihood values) have been used to track occluding objects from multiple cameras [5]. Similarly, silhouettes have also been used for 3-D voxel reconstruction [6] but the complexity of these methods increases linearly in the number of cameras. 3-D voxels have been used to recognize an activity in a scene [7], to estimate pose and register the body parts [8]. In our work, we utilize the inherent sparsity present in background subtracted silhouette images from multiple cameras to localize and track objects in observation region. We extend the method to reconstruct 3-D voxels using silhouette images from multiple cameras. Due to the finite resolution of our observation region over which we estimate our object parameters we show that our method is scalable with the number of cameras.

The paper is structured as follows. In Section 2, we briefly describe the concept of compressed sensing. In Section 3, we formulate the problem and present the method for tracking and reconstruction. Experimental results using real data are presented in Section 4.

Email: {dikpal, aswch, rama}@umiacs.umd.edu, {volkan}@rice.edu

2. COMPRESSED SENSING

2.1. Sparsity

Compressed sensing [1, 2] exploits the structure present in signals using sparse representations with efficient methods for reconstruction. Many signals are sparse in some transformation space. Let $\boldsymbol{f} \in \mathbb{R}^N$ be sparse in some basis $\boldsymbol{\Psi}$ — i.e.

$$\boldsymbol{f} = \boldsymbol{\Psi}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$. We assume that $\boldsymbol{x}$ is K-sparse, $K \ll N$ — i.e. $\|\boldsymbol{x}\|_0 \leq K$ where $\|\boldsymbol{x}\|_0$ is the number of non-zero components in $\boldsymbol{x}$. The central result in CS says that $\boldsymbol{x}$ can be recovered exactly from $M << N$ number of non-adaptive linear projections of $\boldsymbol{f}$. Since $M << N$ the recovery of $\boldsymbol{x}$ is ill-posed but the condition of sparsity of $\boldsymbol{x}$ makes recovery possible. The theory [3] states that if the measurement basis $\boldsymbol{\Phi}$ is incoherent with the transformation basis $\boldsymbol{\Psi}$ then we can perfectly recover the signal coefficients $\boldsymbol{x}$. We can recover $\boldsymbol{x}$ using

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}, \tag{2}$$

where $\boldsymbol{\Phi}$ is a $M \times N$ random measurement matrix and

$$M = \mathcal{O}\left(K \log\left(\frac{N}{K}\right)\right). \tag{3}$$

A randomly generated matrix $\boldsymbol{\Phi}$ with IID Gaussian or Bernoulli/ Rademacher ± 1 vectors is incoherent with high probability to an arbitrary fixed basis.

2.2. Recovery

It is possible to recover K-sparse $\boldsymbol{x}$ by performing a combinatorial search over $\binom{N}{K}$ possible sparse subspaces. But, this problem is NP-hard. Due to the incoherency of the basis the following linear program [3] yields the same solution as the combinatorial search

$$\widehat{\boldsymbol{x}} = \arg\min \|\boldsymbol{x}\|_1 \quad \text{s. t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}. \tag{4}$$

This optimization problem, called *Basis Pursuit*, can be solved using linear programming techniques in polynomial time. This yields ℓ_1-sparse coefficients which are equivalent to the desired ℓ_0-sparse coefficients. When the observed signal $\boldsymbol{y}$ has a small error term $\boldsymbol{e}$, deterministic or stochastic, with $\|\boldsymbol{e}\|_2 \leq \epsilon$ the following model holds

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x} + \boldsymbol{e}. \tag{5}$$

In this case the reconstruction program [4] takes the form of

$$\widehat{\boldsymbol{x}} = \arg\min \|\boldsymbol{x}\|_1 \quad \text{s. t. } \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}\|_2 \leq \epsilon. \tag{6}$$

This program is an instance of second order cone programming (SOCP) and results in a unique convex solution.

3. PROBLEM DESCRIPTION

In computer vision, silhouette images are used for various applications like tracking, activity recognition, building 3-D models using voxels etc. Silhouette images can be considered as sparse matrices where few pixels are in the foreground and most in the background. The sparsity of the silhouette images corresponds to the sparsity of object parameters. We utilize the sparsity of silhouette images, using their compressed samples, to directly recover the object parameters in a multi-view setting. We first formulate the multi-view tracking problem and then consider the formulation of 3-D voxel reconstruction task. We assume that multiple cameras are observing a scene. A common approach involves all the cameras sending the data to a central location where it is used to detect and track objects and look out for abnormal activities. We show that for the purpose of tracking it is sufficient to send the random projections of the silhouette image vectors obtained from background subtraction, locally computed at the cameras.

Suppose we have the observation region $\boldsymbol{O}$(which can be either 2-D or 3-D space), being observed by synchronized cameras $c = 1, ..., C$. We assume that most of the region is visible to all the cameras. At any frame $f \in \{1, 2, ..., F\}$ we have background subtracted silhouette images $\boldsymbol{I}_c^f$ (of size $N_{row} \times N_{col}$) at each of the cameras. The foreground in the image is the moving object which we are interested in tracking and on which we would like to further focus and perform our analysis. The foreground is sparse in the image plane. This implies that in the corresponding observation region $\boldsymbol{O}$, the objects or people corresponding to the image foreground are sparse i.e. the area (or volume) occupied by the moving objects or people is very small compared to the area (or volume) of the observation region. We relate the silhouette image and the corresponding objects in the observation region by a linear transformation. For simplicity, we first consider a 2-D observation region $\boldsymbol{O}$ where camera c is provided with homography $\boldsymbol{H}_c$ between the world plane and the image plane.

Assume that for some frame f the cameras are observing the 2-D observation space $\boldsymbol{O}$ as shown in Fig. 1. The region is divided into non-overlapping, tightly packed subregions $n = 1, ..., N$ where (x_n, y_n) are the coordinates of the representative point (like points on the ground plane in Fig. 1) of the subregion n, known at the cameras. In tracking application we would like to localize the objects to one of the regions. Assume that camera c observes the background subtracted silhouette image $\boldsymbol{I}_c^f$ (foreground has value 1 and background 0) with image coordinates (u, v). We define vectors $\boldsymbol{x}$ and $\boldsymbol{y}_c'$ associated with the object location on the 2-D plane and the silhouette image respectively. $\boldsymbol{x}$ is a $N \times 1$ vector with $\boldsymbol{x}(n) = 1$ if object is present in the subregion n and 0 otherwise. In Fig. 1, $\boldsymbol{x}$ is the indicator function of the objects at points on the ground plane and $\boldsymbol{y}_c'$ is the indicator function of foreground at corresponding points on the silhouette image at camera c. Typically in tracking scenarios $\boldsymbol{x}$ is sparse since the objects of interest occupy a small area in the observation region. For every frame f we would like to know the position of the objects, in other words our desired variable is $\boldsymbol{x}$. But, what the cameras instead observe is the background subtracted silhouette image $\boldsymbol{I}_c^f$ from which they construct the $N \times 1$ vector $\boldsymbol{y}_c'$ — i.e.

$$\boldsymbol{y}_c'(i) = \boldsymbol{I}_c^f(u_i, v_i), \tag{7}$$

where the image coordinates (u_n, v_n) of camera c are related to the coordinate (x_n, y_n) of the representative point n by

$$\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix} \backsim \boldsymbol{H}_c \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}, \tag{8}$$

(it should be noted that since (u_n, v_n) take integer values we round the right hand side). The equation relating $\boldsymbol{x}$ and $\boldsymbol{y}_c'$ is then given by

$$\boldsymbol{y}_c' = \boldsymbol{A}_c\boldsymbol{x}, \tag{9}$$

where $\boldsymbol{A}_c$ is an identity matrix in this setting. Given $\boldsymbol{O}$, each camera can compute $\boldsymbol{y}_c'$ as described above. A simple projection of the silhouette from a single camera on the ground plane gives an estimate of the object location but this is not accurate since the parts of the object in parallax do not register under the homography projections. To accurately estimate the position of the objects on the ground plane from $\boldsymbol{y}_c'$, information from multiple cameras is used. For this, cameras need to transmit the information to a centralized location where the computation can be performed. Noting that $\boldsymbol{x}$ is sparse we can significantly decrease the amount of data to be transmitted to the central processing center by projecting the signals into lower dimensions and recovering it using the principle of CS. We assume the vector $\boldsymbol{x}$ to be K-sparse. This means that we can randomly project the vector $\boldsymbol{y}_c'$ into lower dimensions using the $M \times N$ projection matrix $\boldsymbol{\Phi}_c$ with entries from Gaussian distribution $\mathcal{N}(0, 1/N)$ where the number of rows M of $\boldsymbol{\Phi}_c$ is given by (3). Errors are introduced in the signal $\boldsymbol{y}_c'$ (and hence in $\boldsymbol{\Phi}_c\boldsymbol{y}_c'$) due to errors in silhouettes, rounding off errors in (8) and when not all subregions in $\boldsymbol{O}$ are visible to camera c. We model the errors in $\boldsymbol{\Phi}_c\boldsymbol{y}_c'$ as additive white Gaussian(AWG) noise. The resulting equation is

$$\boldsymbol{y}_c = \boldsymbol{\Phi}_c\boldsymbol{y}_c' + \boldsymbol{e}_c. \tag{10}$$

The cameras transmit $\boldsymbol{y}_c$ to the central location where they are stacked to form vector $\boldsymbol{y}$

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \\ \vdots \\ \boldsymbol{\Phi}_C \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_C \end{bmatrix} \tag{11}$$

resulting in

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{e} \tag{12}$$

The vector $\boldsymbol{x}$ is recovered by solving (6). For any given frame, after compressing we need to send CM values compared to $CN_{row}N_{col}$ without compression. Also, the recovery of $\boldsymbol{x}$ is a function of N allowing the algorithm to scale in number of cameras. For simplicity, we consider rectangular subregions $n = 1, ..., N$, with the representative points forming a $N_1 \times N_2$ rectangular grid.The problem described above easily extends to 3-D voxel reconstruction. We assume

a 3-D $\boldsymbol{O}$ being observed by C cameras. At frame f, the cameras observe silhouette images $\boldsymbol{I}_c^f$. We assume that at camera c, the projection matrix $\boldsymbol{P}_c$ is known. Unlike multi-view ground plane tracking, in 3-D voxel reconstruction we need to recover all the three coordinates of the object to reconstruct the 3-D shape. We divide the 3-D observation region into N sufficiently dense subregions which are non-overlapping and tightly packed. Here the representative point of the subregion n has coordinates (x_n, y_n, z_n). Again, for simplicity we assume that the subregions are cuboidal volumes called voxels and the representative points form a $N_1 \times N_2 \times N_3$ grid. The subregions are denser compared to tracking and the object occupies a lot more sub-regions than it did in tracking scenario. Similarly, we define $N \times 1$ vectors $\boldsymbol{x}$ and $\boldsymbol{y}_c'$. $\boldsymbol{x}(n) = 1$ if object occupies subregion n and 0 otherwise. Obviously, if $\boldsymbol{x}(i) = 1$ we would have $\boldsymbol{y}_c'(i) = \boldsymbol{I}_c^f(u_i, v_i) = 1$ where

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{P}_c \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \tag{13}$$

and (x_i, y_i, z_i) are the coordinates of the grid point in voxel i. All the voxels whose projection onto the image plane of camera c intersects with the silhouette of image $\boldsymbol{I}_c^f$ are assigned to be occupied — i.e. $\boldsymbol{y}_c'(i) = 1$. Thus for any camera the number of voxels assigned as occupied is greater than the number of truly occupied voxels. Hence, for finding the true voxel occupation $\boldsymbol{x}$ we use silhouettes from multiple cameras. In the region $\boldsymbol{O}$ the volume occupied by the object is assumed to be sparse implying a sparse $\boldsymbol{x}$. To recover $\boldsymbol{x}$ from $\boldsymbol{y}_c'$, $c = 1, \ldots, C$ we follow exactly the recovery procedure adopted in multi-view tracking.

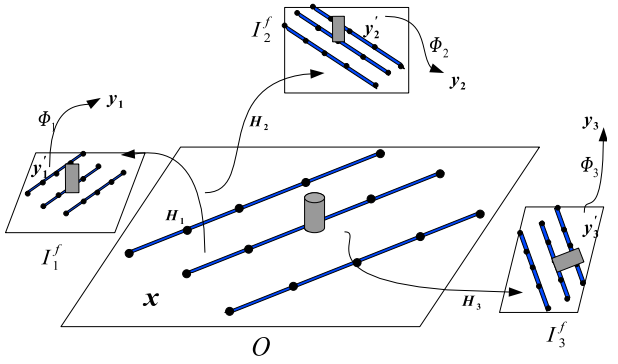

Fig. 1. Ground plane tracking scenario. Observation region $\boldsymbol{O}$ observed by 3 cameras. The points on $\boldsymbol{O}$ are the representative points of the subregions known at cameras. The homography $\boldsymbol{H}_c$ is provided at camera c. $\boldsymbol{I}_c^f$ are the silhouette images at camera c at frame f. The silhouette values at points on image $\boldsymbol{I}_c^f$ form $\boldsymbol{y}_c'$. The sparsity of the silhouette image corresponds to the sparsity of the object(cylinder)

4. RESULTS

4.1. Multi-view tracking

We present the results of an experiment performed using video sequences collected by four cameras located in an outdoor area. The background-subtracted images at the cameras are of size 240×320 ($N_{row} = 240, N_{col} = 320$). First, we detect the objects in the scene and then track them over 400 frames. During detection the observation region $\boldsymbol{O}$ is a rectangular region $60ft \times 55ft$ most of which is observed by all the 4 cameras. We place a sufficiently dense 101×101 ($N_1 = 101, N_2 = 101$) uniformly spaced grid on this region, implying $N = N_1 N_2 = 101^2$ subregions over which we detect and localize the objects. Following the procedure described in Section 3 we recover the vector $\boldsymbol{x}$ which indicates which sub-regions have the object. We detect 2 objects which we track over the next 400 frames. Since each object occupies more than a point, instead of recovering an exact 2-sparse vector $\boldsymbol{x}$ we get a more dense vector. The location is estimated by averaging over these 2 dense blobs. Once we detect the objects, we track them using a similar procedure but for tracking we confine our region of search to a rectangular region of size $20ft \times 20ft$ centered at the detected object locations. For tracking, the observation region $\boldsymbol{O}$ at frame f is centered around the object location estimated at frame $f-1$. On this we place a grid of size 26×26 ($N_1 = 26, N_2 = 26$) where unlike detection the grid points are distributed according to a Gaussian distribution centered at the object location and with variance $3.5ft$ in both directions. A Gaussian spaced grid allows us to account for the expected small movements as well as the large ones. It also decreases the complexity by decreasing N. The observed vector is randomly projected using a matrix with IID Gaussian entries. We add additive white Gaussian noise with SNR $= 5$dB. The tracking results are shown in Fig. 2

We use the software l_1-Magic [9] to perform Basis Pursuit with quadratic constraint. In practice the algorithm has a complexity of $\mathcal{O}(N^p)$ where typically $3 \leq p \leq 4$ and $N = N_1 N_2$ is the size of vector $\boldsymbol{x}$. This makes the complexity $\mathcal{O}(N_1^p N_2^p)$ of the recovery algorithm prohibitively huge. Also, it demands tremendous amounts of memory. Hence, we divide the grid ($\boldsymbol{O}$) into smaller chunks ($\boldsymbol{O}_1, \ldots, \boldsymbol{O}_{N_1}$) for the purposes of random projection and recovery as shown in Fig. 1. The lines on the ground plane correspond to the chunks $\boldsymbol{O}_r$. Instead of randomly projecting the entire vector $\boldsymbol{y}_c'$ we use sub-vector $\boldsymbol{y}_{cr}'$ corresponding to the row $\boldsymbol{O}_r$ and reconstruct $\boldsymbol{x}_r$, $r = 1, \ldots, N_1$ — i.e. we consider vectors of size N_2. The idea of processing the grid data line-wise is similar in nature to computer tomography type approaches using the radon transform. Now, we have a complexity of $N_1 \mathcal{O}(N_2^p)$ ($\ll \mathcal{O}(N_1^p N_2^p)$) for recovering entire $\boldsymbol{x}$. We use

$$M = \alpha K \log_2\left(\frac{N}{K}\right), \alpha \gtrapprox 1, \tag{14}$$

for random projections of $\boldsymbol{y}_{cr}'$ where $K \approx \|\boldsymbol{y}_{cr}'\|_0$. With the above method we have an average communication of 353 measurements per camera for the first object and 359 measurements for the second object.

4.2. 3D voxel reconstruction

We performed the 3-D reconstruction experiment in an indoor setting for one frame where the object is being observed by $C = 8$ cameras placed around it. The background-subtracted silhouette images are of size 484×648. The observation region is a $0.8m \times 0.82m \times 1.62m$ space. We place a sufficiently dense uniformly space grid of size $81 \times 83 \times 163$ ($(N_1 = 81, N_2 = 83, N_3 = 163)$) in this region. As in the tracking scenario we randomly project the observed vector using matrix with IID Gaussian entries and add additive white Gaussian noise with SNR $= 5$dB to account for the errors in the observed vector. To decrease the complexity of the recovery algorithm we divide the grid into smaller chunks of size N_1 along one of the rows. We recover $\boldsymbol{x}$ which has values close to 1 corresponding to the region occupied by the object and lower values in the empty regions.

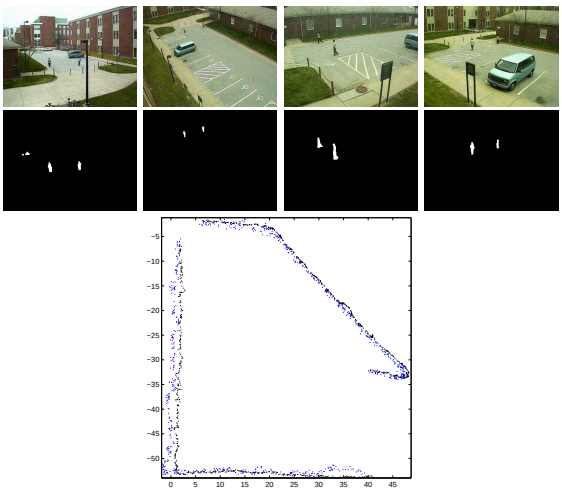

Fig. 2. Outdoor scene of size $60ft \times 55ft$ observed by $C = 4$ cameras. Tracking results on a video sequence of 400 frames. The first two rows show sample images and background subtracted silhouettes respectively. These background subtracted silhouettes are used to track objects on the ground plane. The bottom image shows the tracked points (blue) as well as the ground truth (black).

The reconstructed 3-D voxels shown in Fig. 3 were obtained by thresholding $\boldsymbol{x}$ at 0.7 and displaying voxels with high values. The voxels corresponding to the object were downsampled 3 times in each direction for the sake of display. We can see that our method of reconstructing 3D-voxels is robust to errors in the silhouette images.

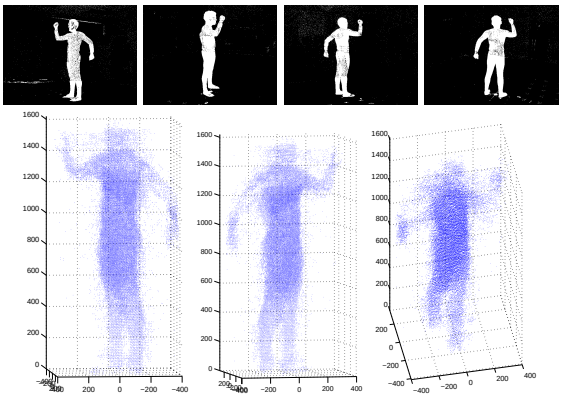

Fig. 3. Indoor scene of size $0.8m \times 0.82m \times 1.62m$ overlayed with grid of size $81 \times 83 \times 163$ observed by $C = 8$ cameras. (Top) Background subtracted silhouette images . (Bottom) Three views of the reconstructed object.

4.3. Scalability

We performed an experiment on synthetic data to show that our method scales with the number of cameras. We placed a sphere of unit size at the origin and placed C virtual cameras at 10 units from origin observing the sphere. To compare our method, we also reconstructed the sphere using the 3-D voxel reconstruction method described in [6]. We estimated the time taken to reconstruct the sphere from C cameras as C was varied from 5 to 30. We plot the relative computational time(RCT) achieved by both the methods in Fig. 4. We define RCT as the ratio of time taken by C cameras to time taken by 5 cameras. Our method has a relatively flat curve compared to linearly increasing one for the other method. This result is expected since our method is scalable with cameras since it is dependent only on the resolution of the observation space. The same holds for ground-plane tracking when compared to [5].

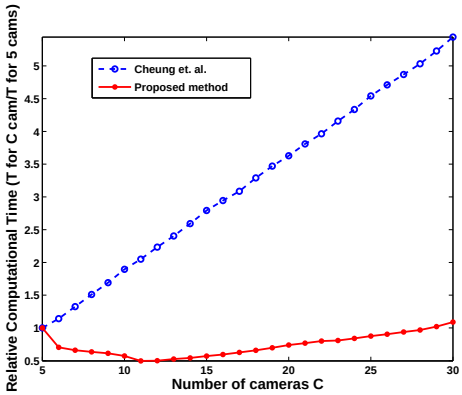

Fig. 4. Speedup achieved by our method and by Cheung et. al. [6]

5. CONCLUSIONS

We have proposed an approach for two multi-view estimation problems by applying the theory of CS to sparse, background-subtracted silhouette images and showed that our approach is scalable with the number of cameras. We utilized the sparsity in silhouettes and hence in object parameters to track objects from multiple cameras using very few measurements. We extended the method to 3-D voxel reconstruction. The results show that we can exploit the sparsity of silhouettes to directly recover the object parameters like location and shape. A future direction of work would be to extend our method from silhouettes to complete images to fully utilize the compression offered by the proposed method.

6. REFERENCES

[1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[4] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[5] S. M. Khan and M. Shah, "A multi-view approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*, 2006, vol. 4, pp. 133–146.

[6] G. K. M. Cheung, T. Kanade, J. Y. Bouget, and M. Holler, "Real time system for robust 3D voxel reconstruction of human motions," in *CVPR*, 2000, pp. 714–720.

[7] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *International Conference on Computer Vision*, 2007, pp. 1–7.

[8] A. Sundaresan and R. Chellappa, "Segmentation and probabilistic registration of articulated body models," in *International Conference on Pattern Recognition*, 2006, vol. 2, pp. 92–96.

[9] E. Candès and J. Romberg, "L1-magic," `http://www.l1-magic.org/`, 2007.