Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Adaptive compressive sensing algorithm for video acquisition using a single-pixel camera

Imama Noor
Eddie L. Jacobs

SPIE

IS&T
imaging.org

Adaptive compressive sensing algorithm for video acquisition using a single-pixel camera

Imama Noor
Eddie L. Jacobs
University of Memphis
Department of Electrical and Computer Engineering
206 Engineering Science Building
Memphis, Tennessee 38152
E-mail: inoor@memphis.edu

Abstract. *We propose a method to acquire compressed measurements for efficient video reconstruction using a single-pixel camera. The method is suitable for implementation using a single-pixel detector, along with a digital micromirror device or other types of spatial light modulators. Conventional implementations of single-pixel cameras are able to spatially compress the signal, but the compressed measurements make it difficult to exploit temporal redundancies directly. Moreover, a single-pixel camera needs to make measurements in a sequential manner before the scene changes, making it inefficient for video imaging. We discuss a measurement scheme that exploits sparsity along the time axis for video imaging. After acquiring all measurements required for the first frame, measurements are acquired only from the areas that change in subsequent frames. We segment the first frame, detect the magnitude and direction of change for each segment, and acquire compressed measurements for the changing segments in the predicted direction. Next, we compare the reconstruction results for a few test sequences with existing techniques and demonstrate the practical utility of the scheme.* *[DOI: 10.1117/1.JEI.22.2.021013]*

1 Introduction

The compressed sensing (CS) framework for image acquisition exploits the inherent properties of a signal to reduce the number of samples required for reconstruction. Most signals are sparse in some domains and require fewer samples than specified by the Nyquist criterion to fully recover the signal. Unlike conventional Nyquist sampling, CS uses the sparsity information in the signal and acquires measurements in the domain in which the signal is sparse. The sampling matrices or projections are carefully designed to acquire maximum information from the signal. Random projections have been proven to recover the signal with the least number of samples above the minimum bounds, and with high probability. Some deterministic sampling matrices have also been investigated and proven efficient for a full recovery of the signal.

Researchers have come up with many imaging architectures for CS implementation employing spatial light modulators (SLMs). These include the Rice single-pixel camera, which uses a digital mirror device (DMD),[1] coded apertures, and CMOS SLMs[2] to exploit the sparsity of the signal in the spatial domain.[3] These architectures acquire compressive measurements in the spatial domain, but due to the sequential nature of measurement acquisition, they are not efficient for video. Most CS architectures use a single detector that creates a temporal bottleneck for applications requiring fast sampling rates. One way to reduce the effect of this bottleneck is to take multiple measurements at one instance by increasing the number of sensors. Each sensor will require a respective DMD, making an array of DMDs that is not a feasible solution in terms of cost and space. Another cost-efficient way is to exploit redundancies or its complement, temporal sparsity, in a video sequence.

In order to exploit temporal sparsity in a video, we intuitively think of changes between frames. In many video sequences, change is sparse along the time axis. Many methods have been published for video sensing and reconstruction that exploit change sparsity, most of which acquire measurements for several frames and reconstruct them subsequently using Fourier or wavelet domain sparsity. One direct method is to acquire measurements for each frame separately. To minimize motion blur, direct acquisition requires the scene to be static before measurements are made for each frame, which is not practical in most cases. Another approach adopted is three-dimensional wavelet reconstruction.[1] Samples for a group of frames are acquired and a wavelet basis is used for recovering all frames in the group at once. This method cannot be used for real time video streaming without incurring latency and delay that may significantly affect performance in many situations. Frame differencing has been used where the differences between consecutive frames are compressively measured, reconstructed, and added to the previous frame.[4] This method not only accumulates residual error, but the mean square error (MSE) increases significantly when the difference is not sparse, as in the case of large changes in the scene. Another approach is based on modeling specific video sequence evolution as a linear dynamical system (LDS).[5] This approach reduces the required samples for reconstruction considerably but is restricted to videos possessing an LDS representation, which is possible for only a few specific sequences. Some works based on block-based compressive sensing (CS), such as block-based compressive

Paper 12314SSP received Aug. 16, 2012; revised manuscript received Mar. 13, 2013; accepted for publication Mar. 22, 2013; published online May 7, 2013.

sensing with smooth projection Landweber reconstruction (BCS-SPL), divide the frame into nonoverlapping blocks and process each block separately. The basic technique splits the processing into smaller blocks and combines the reconstruction for the final result.[6] This method does not take into account the temporal redundancies in a video. More advanced techniques based on BCS-SPL take into account motion estimation parameters to aid the reconstruction process. Motion estimation/motion compensation BCS (ME/MS-BCS) selects a group of pictures (GOP) for estimating motion vectors and reconstruct the GOP using this information. This improves the subrate performance but incurs an undesirable time delay in addition to increasing reconstruction complexity.[7–10] One other approach is adaptive block-based compressive sensing in which a frame is divided into a specific number of blocks and each block is assigned measurements based on changes and texture.[11] This approach accumulates residual error and gives block artifacts after recovery. Moreover, it is computationally expensive to optimize measurement allocation before each frame acquisition.

In this study, we try to exploit temporal redundancies and reduce the required computations. We propose a scheme keeping in view the properties of a single-pixel camera employing a DMD. Takhar has shown an implementation of CS using a single-pixel detector for sensing light after modulation from a DMD.[1] This setup requires the scene to be static before all the required measurements are made. However, by using the concept of sparse changes along the time axis, we can reduce the number of samples required for reconstruction of each frame. Most real video frames are defined by contours and moving objects trajectories are irregular in shape. The scheme that we propose identifies static and dynamic regions of arbitrary shape for each frame and only modulates the light incident from these identified dynamic locations for each frame. The spatial modulation scheme is random for identified dynamic regions and allows recovery of the signal with fewer samples than Nyquist[12] reconstruction. In order to achieve video rates, the dynamic and static region identification should be fast enough to allow detection of these regions and the subsequent measurements to take place within a specific time frame. We have used optical flow features to complete this task. The next step of the scheme uses features of the moving segments to predict the direction of motion and allocate additional measurements in that direction. It confines the measurements to the areas predicted to change over time. The change detection is performed at the sensor, with the result that predictions are faster to make. The mask formed by these dynamic areas reduces the number of pixels to be estimated, hence reducing the load on the reconstruction algorithm, which allows for faster reconstruction.

The scheme uses the multiplexing property of SLMs in imaging sensor design. Most DMD or other SLMs can selectively direct light from individual image pixels toward the detector. Using the dynamic region detection scheme described earlier, the flux of light from dynamic regions in the scene is multiplied by random projections, the results of which are then summed at the detector. The number of distinct random patterns to generate measurements for reconstruction depends on the size and sparsity of the dynamic region. The algorithm also measures the direction in which a particular region is progressing, thereby spreading the region of measurement in that direction. Light spatially modulated by calculated projections is sampled at less than Nyquist rates for a full recovery. The patterns used for modulation are critical to noiseless recovery. The projecting matrices follow the restricted isometry property (RIP).[13] We give some background of the fundamental concepts used for this technique in Sec. 2. Section 3 details all aspects of the scheme. Section 4 discusses the simulation results, and Sec. 5 summarizes the conclusions and future work needed to improve on this method.

The most effective application of this scheme is for videos with slow changes over time or few spatially dynamic regions. It reduces the computation required after each frame, gives comparable peak signal-to-noise ratio (PSNR) with other techniques with the least number of measurements, and is implementable on a single-pixel DMD camera.

2 Theoretical Background

2.1 *Compressed Sensing*

CS provides an efficient framework to sample a signal below the Nyquist rate without losing high-frequency information. In order to use the CS framework for acquisition, some conditions must be met. These conditions are based on the characteristics of the signal under consideration, and they guarantee a full recovery from far fewer samples than conventional methods require. Consider a signal x, which we desire to reconstruct from the measurements y. The relationship between the signal and the measurements can be expressed as

$$y = Ax + \omega, \tag{1}$$

where y is defined as the measurement vector of length k, x is the signal to be recovered having n dimensions, the matrix A is the measurement matrix, and w is the noise with an assumed Gaussian distribution $(0, \sigma I)$. This is an underdetermined system of linear equations where $k = n$. If the signal is sparse in a domain and the sparse domain is incoherent with the sampling domain, then we should be able to represent the signal well by a linear combination of only a few atoms of the domain and fully recover the signal. For instance, if x is sparse in the domain ϕ then

$$x = \phi z \tag{2}$$

and

$$y = A\phi z + \omega, \tag{3}$$

where the vector z is a sparse vector with a very low sparsity index. The sparsity index is the number of nonzero elements in a coefficient vector. CS theory provides a threshold imposed on the sparsity index based on mathematical derivations. If S is the sparsity index of x in ϕ, then the aspect ratio of the measurement matrix required for full recovery is given by the ratio

$$\frac{k}{n} \geq \frac{CS\log(n)}{n}, \tag{4}$$

where C is a positive constant. According to Candès,[13] C depends on the coherence measure of the measurement matrix and the sparsity basis. If a_i and ψ_j represent the atoms of A and ϕ, respectively, then

$$\mu = \max_{1\leq i,j\leq n} |\langle a_i, \psi_j \rangle| \tag{5}$$

and the matrices are incoherent if the following holds:

$$n^{1/2} \leq \mu \leq 1. \tag{6}$$

In addition to satisfying the coherence measure, the sampling matrix should comply with the RIP, which requires that if a matrix operates on a sparse vector, the l_2 norm of the resulting product should be less than δ_s which is a restricted isometry constant.[13] If z is a sparse vector, then the product $A\phi$ should satisfy this equation:

$$(1-\delta_s)\|z\|_2^2 \leq \|A\phi z\|_2^2 \leq (1+\delta_s)\|z\|_2^2. \tag{7}$$

For a given matrix, determining whether it satisfies RIP is an NP hard problem.[14] The RIP ensures that the transformation $A\phi$ preserves the distances between the nonzero planes of sparse vectors. This is equivalent to the requirement that the largest eigenvalue of $A\phi(A\phi)^T$ lies in the interval $[1+\delta_s, 1-\delta_s]$. Most random matrices satisfy both the incoherence and RIP criteria. Some deterministic matrices exhibit statistical RIP compliance and fully recover the signal with a high probability.[15]

A signal acquired using a sampling matrix satisfying the above mentioned criteria can be recovered by using CS reconstruction algorithms for underdetermined systems of linear equations. It has been established that samples acquired using a measurement matrix satisfying k-neighborliness can be recovered, provided that the solution is sparse.[16] Different optimization techniques such as basis pursuit (BP) can be used for estimating a solution.[17,18] We formulate the problem as a BP problem and optimize it using total variation minimization.[19–21] According to CS theory, if we minimize the l_1 norm, it gives us the sparsest unique solution to this convex problem:[22]

$$\text{minimize}\|x\|_1 \quad \text{subject to } y = Ax. \tag{8}$$

If we decompose x into ϕz and minimize the l_1 norm subject to the constraints described next, then

$$\text{minimize}\|z\|_1 \quad \text{subject to } y = A\phi z. \tag{9}$$

If the signal is sparse spatially, ϕ can be the identity matrix. In the case of noisy measurements, we formulate the problem as BP denoising or[23]

$$\text{minimize}\|x\|_1 \quad \text{subject to } \|y - Ax\|_2 < \varepsilon. \tag{10}$$

If $x = fz$, then

$$\text{minimize}\|z\|_1 \quad \text{subject to } \|y - A\phi z\|_2 < \varepsilon, \tag{11}$$

where $\varepsilon > \sigma$ and σ is the combined noise variance due to detection and quantization noise. Detector noise depends on the integration time, and by increasing integration time, this error can be reduced. In order to minimize quantization noise, it is possible to make a trade-off between saturation and quantization while using compressive projections for measurements.[24] Since each compressive measurement carries the same information, by adjusting the quantization levels to saturate more measurements, the quantization error can be greatly reduced on all the unsaturated measurements.

2.2 *Segmentation and Change Detection*

Segmentation is a key step for implementing our proposed scheme. It separates the scene into segments, which helps to identify and track object motion effectively. In this study, we used normalized cut image segmentation.[25] The effectiveness of the technique varies depending on the contrast and texture in a video frame. The main goal here is to effectively separate portions of the image that may move in subsequent frames from static background portions of the scene. An important parameter is the number of segments produced during segmentation. This can vary depending on the sparsity of a particular video sequence to achieve optimum segmentation. In order to determine the number of segments, we maximize the inter-cluster distance and minimize the intra-cluster distances over a predefined range of number of clusters. The contrast measure between clusters can be maximized to separate a cluster from neighboring clusters.[26] In order to decrease intra-cluster distances, variance can be minimized. We find the number of segments corresponding to a maximum value of contrast by searching over a range of the number of segments. This range is bounded by the number of segments corresponding to the value of mean variance and a specified maximum number of segments. Variance for l clusters can be calculated as

$$\sigma = \frac{1}{l}\sum_{i=1}^{l}\sigma_i = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{m_i}\sum_{k=1}^{m_i}|x_k - \mu_i|, \tag{12}$$

where σ_i is the variance of pixels within the segment i and m_i is the mean of the i'th segment, l is the total number of segments and m_i corresponds to the number of pixels within the segment i. Mean contrast for l clusters can be computed using the equation here:

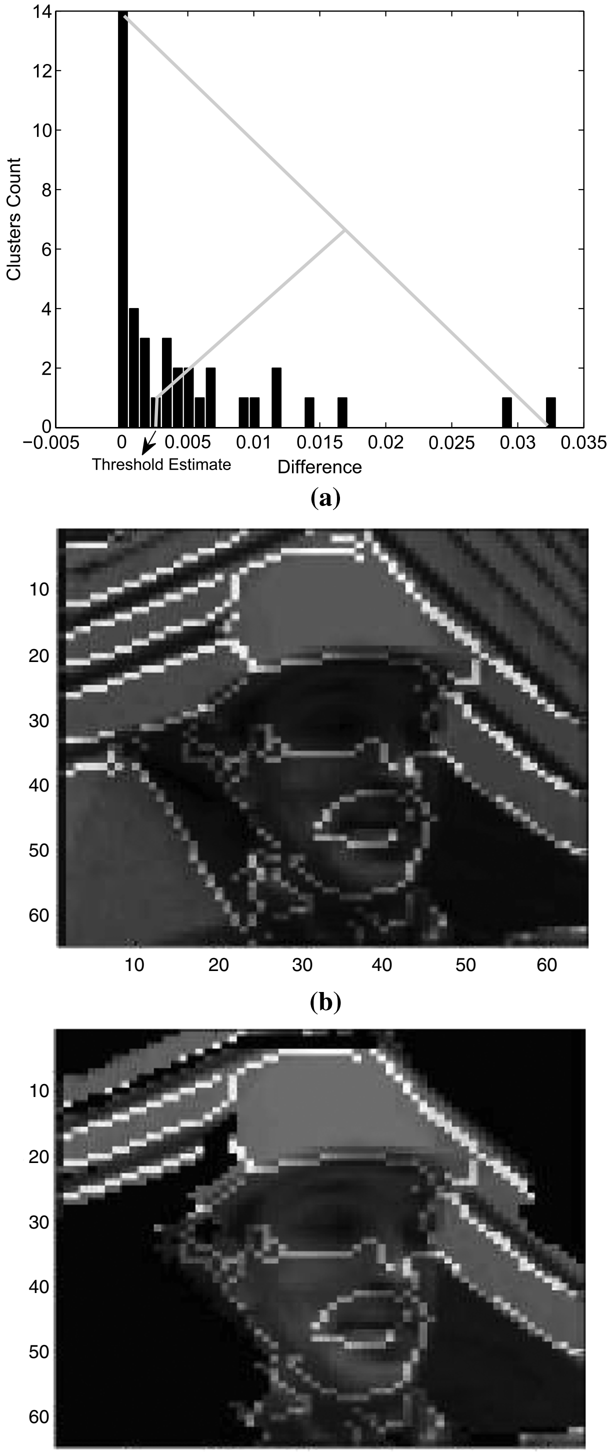
$$Ct = \frac{1}{l}\sum_{i=1}^{l} Ct_i = \frac{1}{l(l-1)}\sum_{i=1}^{l}\sum_{j=1}^{l}\sqrt{\frac{(\mu_i-\mu_j)^2+(\sigma_i-\sigma_j)^2}{(\mu_i+\mu_j)^2+(\sigma_i+\sigma_j)^2}} \quad j \neq i, \tag{13}$$

where Ct is the mean contrast of all segments. The variation of the number of segments against maximum contrast is used to determine the number of segments used for all further processing.

We next classify segments as static or dynamic. The static and dynamic areas are found by looking at the temporal differences in the average pixel value over a segment. In the context of a single-pixel DMD-based sensor, averages over arbitrary-shaped regions are easily calculated by simply directing the light associated with those pixels to the single detector and dividing the measured value by the number of pixels in the segment. These averages are compared with a threshold value. We assume that only a few segments change significantly from one frame to the next. Therefore, the histogram of differences between the segment averages forms a unimodal distribution with a mode at the first bin. We estimate the change detection threshold based on unimodal distribution to select segments that have significantly changed between consecutive frames. A straight line joining peak and last bin is drawn as shown in Fig. 1(a). The value corresponding to the point on the histogram with maximum divergence from the straight line gives an estimate of the threshold.[27] The area above the threshold is classified as dynamic, and the area below is static. In the event that more segments change significantly, the histogram is inverted to calculate the threshold. All segments are selected in the case where no population is considerably larger than the other. The practical implementation of this procedure is described in Sec. 3, later in this article.

2.3 *Motion Estimation*

In order to sense a video efficiently, we need to perform both spatial and temporal compression. Random sampling can compress the signal effectively if applied in the domain where the signal is sparse. In the time domain, changes are sparse in many video sequences. In order to take measurements efficiently in the change domain, we need a transformation that detects change. Most transformations require knowledge of past and future frames to calculate the change. This approach can compress the signal after acquisition or by using additional hardware to reach video rates. Another way to detect change is predictive modeling, but due to the ephemeral nature of videos, it is hard to model these changes, with the exception of a few specific video sequences. In this study, a simple optical flow technique is implemented, which uses the multiplexing properties of DMDs employed in single-pixel cameras.[1] The scheme locates the dynamic and static segments in a frame and takes some strategic averages to access the direction in which the segments might be moving. The scheme predicts the direction of motion based on a novel feature of moving segments and allocates measurements in that direction for the next frame. The process begins by encircling each segment by a circle with a radius larger than the maximum size of the segment. The circle is divided into sectors. The difference in averages outside the segment and inside the circle generates peaks at values of the angle θ, which predicts the direction in which the segment is moving.[28] The possible directions are quantized by the number of sectors used to divide the circle (eight in the current research). It is assumed that motion is sparse along the time axis and only a few segments move between consecutive frames. A binary filter is formed based on these locations, which filters out the static segments and enables measurements to be performed for only the dynamic pixels of moving objects. The binary pattern and the random sampling matrix are multiplied to acquire samples.



(a)

(b)

(c)

Fig. 1 (a) Threshold estimation. (b) Segmented frame. (c) Selected segments with differences above the threshold.

2.4 *Hardware Implementation*

The proposed method can be efficiently realized in hardware. The basic operations to implement this scheme can be computed rapidly. To acquire measurements for dynamic regions, the mirrors of the DMD are turned off where the mask is zero, and a random pattern is projected on the rest of the mirrors. The sampling requirements for the discussed algorithm depend upon the dynamic region in a video sequence and varies directly with the number of pixels that are changing. Our proposed technique adapts the sampling to these changing regions. In order to achieve real-time video streaming, the sampling process on the encoder side would need to be fast enough to acquire dynamic measurements in less than 33 ms. On the decoder end, the computations would need to be fast enough to optimize the dynamic area in less than 33 ms. Under the assumption of slow change, the segmentation process can be performed in parallel and the newest segmentation can be used to form sampling masks instead of requiring the latest frame to be reconstructed. This saves significantly on reconstruction time. The discussed scheme for measurement allocation is capable of acquiring a number of frames without consuming as many resources for many types of video compared to existing techniques. The bandwidth gained by reducing samples required for a frame reconstruction can be used toward increasing frame rate or decreasing bandwidth.

3 Methodology

The setup described by Takhar[1] is used as an example of practical implementation (see Fig. 2). We propose to acquire all the measurements for the reconstruction of the first frame by projecting different random patterns on the entire DMD. For subsequent frames, only temporally changing regions of the image are measured. To calculate magnitude and direction of change, we adopt the scheme shown in the flowchart in Fig. 3.

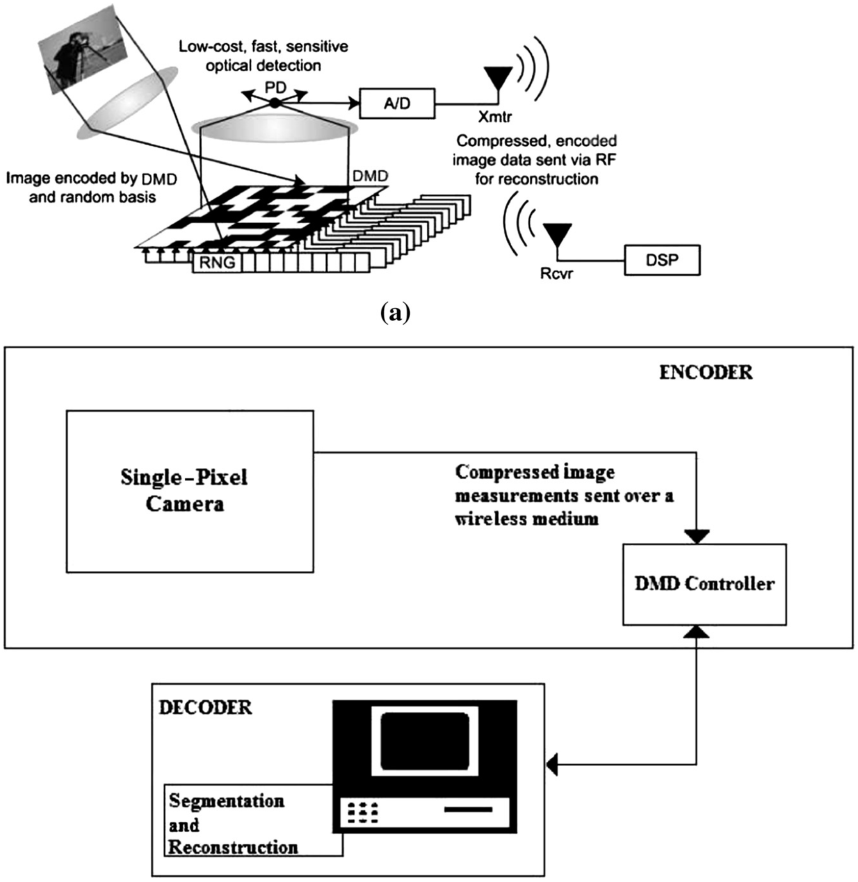
We segment the first frame after reconstruction into a number of segments. The segmentation algorithm used is based on a normalized cut criterion, which measures both dissimilarity between the groups and similarity within a group. It maximizes the normalized cut criterion for a given number of clusters. In order to estimate change due to each moving object between consecutive frames, we try to separate each object from the background and estimate the minimum number of groups that can separate objects present in an image.

In order to optimize the number of segments, we maximize the contrast between the segments and minimize the variance within segments in a frame. We start with a minimum of five segments and calculate variance and contrast between each frame. The number of segments is incremented in each iteration until a maximum contrast limit is reached. The contrast is estimated over a range of variance above the mean value, as shown in Fig. 4. The number of segments corresponding to the maximum contrast is used for all subsequent calculations. The segment can be expressed formally as

$$y_p(x,y) = \begin{cases} 1 & x, y \in p^{\text{th}} \text{ segment} \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

The segmentation mask for each segment is used to drive the DMD of the single-pixel camera to route light from each segment to the detector one by one. The output of the detector is the average of each segment. A step by step procedure is listed in Algorithm 1.

Next, we calculate the temporal differences in the averages over each segment to see which segment has changed significantly. We assume that change is sparse between two


Low-cost, fast, sensitive optical detection
PD
A/D
Xmtr
Compressed, encoded image data sent via RF for reconstruction
Image encoded by DMD and random basis
DMD
RNG
Rcvr
DSP
ENCODER
Single-Pixel Camera
Compressed image measurements sent over a wireless medium
DMD Controller
DECODER
Segmentation and Reconstruction

(a)

(b)

Fig. 2 (a) Single-pixel camera.[1] (b) Adaptive compressive sensing imager block diagram.

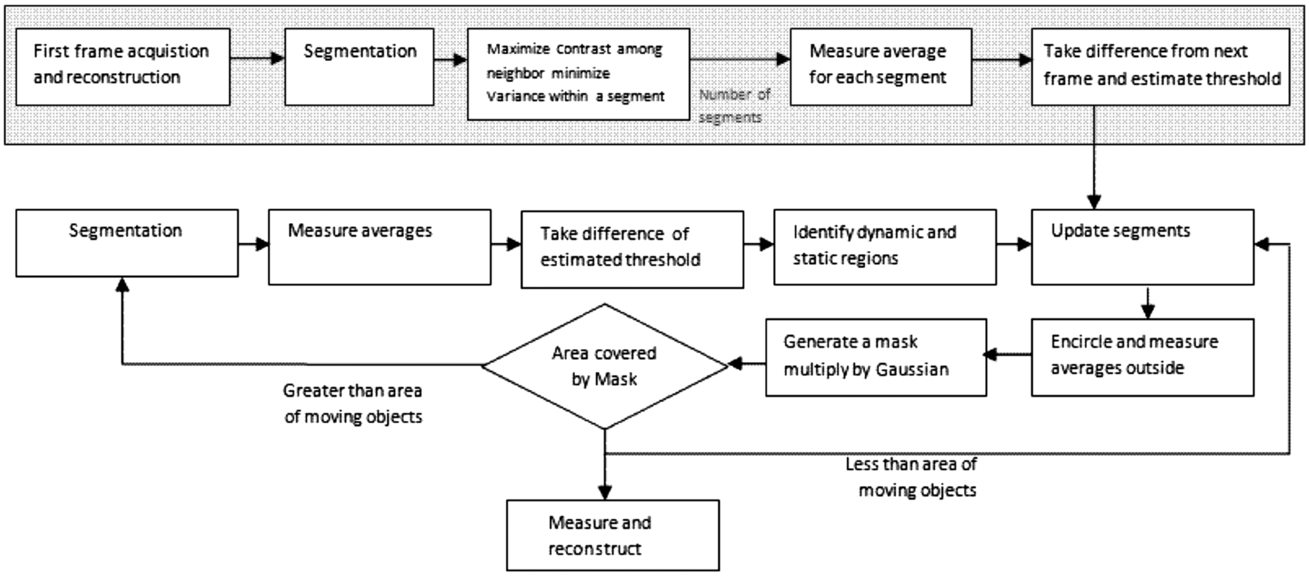
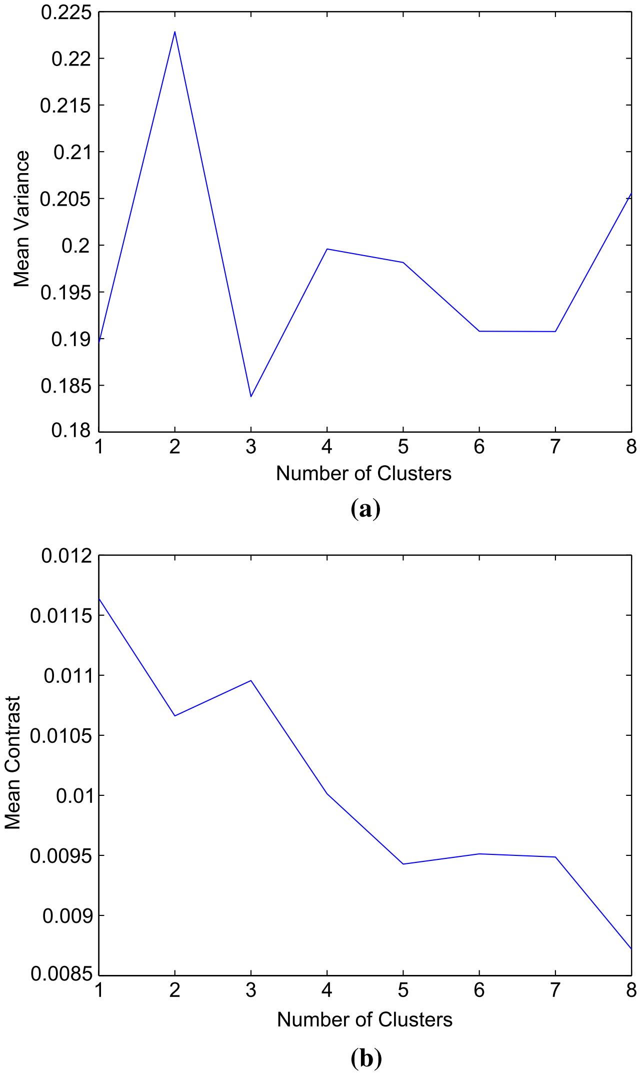

First frame acquistion and reconstruction
Segmentation
Maximize contrast among neighbor minimize Variance within a segment
Number of segments
Measure average for each segment
Take difference from next frame and estimate threshold
Segmentation
Measure averages
Take difference of estimated threshold
Identify dynamic and static regions
Update segments
Area covered by Mask
Generate a mask multiply by Gaussian
Encircle and measure averages outside
Greater than area of moving objects
Less than area of moving objects
Measure and reconstruct

Fig. 3 Flow chart for adaptive sample acquisition.



(a)

(b)

Fig. 4 (a) Mean contrast versus number of clusters. (b) Mean variance versus number of clusters.

consecutive frames and only a few segments change significantly relative to the others. In view of this assumption, unimodal thresholding can be used to estimate the level of significance for change detection.[27] We form the histogram of segment average differences and the threshold is the point of maximum divergence on the curve from the straight line joining the peak and the bin before the last empty bin as shown in Fig. 1. The number of bins is kept equal to the number of segments. Increasing the number of bins does not affect the detection significantly. The probability to detect change accurately depends on the quality of segmentation and the threshold estimation technique. In this approach, if the difference histogram is not unimodal, all segments are selected for further processing. We noticed in our simulations that increasing the number of segments makes the change detection more precise. Considering that the number of segments should be moderate for this scheme to be competitive with other methods, we kept an upper bound of 40 segments with a negligible effect on performance.

The segments with changed averages are selected [see Fig. 1(b) and 1(c)] and encircled with a radius exceeding the distance from the centroid to the farthest pixel in a segment by a fixed amount. For this work, this value was set to 4 pixels. Setting the radius beyond the farthest pixel defines the area within which a segment can move. In order to calculate the magnitude and direction of motion, we partition the circle into 8 equal sectors covering $0 \rightarrow 2\pi$ and representing eight degrees of freedom that a segment can move. For each segment, the space outside the segment boundary and inside the circle is determined. This is projected by the DMD onto the detector to calculate the average in this boundary area. This average is compared with the previous frame average of this same area. We restrict the estimate of the direction of motion to the eight central angles $\theta_k = 1 \ldots 8$ of the eight sectors. We find all directions θ_k for which the difference of the boundary averages exceeds a predefined threshold. Mathematically, the average can be written as

Algorithm 1 Adaptive CS for Video Acquisition Using SPC.

```
1. Acquire the number of measurements for the first frame according to Eq. (4).
2. Divide the frame into a number of segments that maximize contrast and minimize variance within the segments and calculate the vector $\text{SegAvg}_{y_p}(t) = \frac{1}{s_p}\sum_{i,j\in y_p} x_{i,j}$, where $y_p$ is the set of pixel locations in a segment, $p = 1, 2, \ldots, F$, $s_p$ is the number of pixels in $y_p$, and $t$ is the frame number.
3. Measure $\text{SegAvg}_{y_p}(t+1) = \frac{1}{s_f}\sum_{i,j\in y_p} x_{i,j}$ and select $p$, for which $|\text{SegAvg}_{y_p}(t) - \text{SegAvg}_{y_p}(t+1)| > \alpha$, where $\alpha$ is the change threshold determined by unimodal thresholding.
4. Draw a circle $\text{circ}(\text{cen}_{y_p}, \text{rad}_p)$, where $\text{rad}_p$ is greater than the distance between the center and the farthest pixel in the segment.
5. Define sectors $\theta$ from $(k-1)\pi/4 \rightarrow k\pi/4$, where $k = 1, 2, 3, \ldots 8$. Measure each $\text{PartAvg}_{\theta_k}(y_p, t) = \frac{1}{s_p}\sum_{i,j>y_p}^{i,j<\text{rad}_{y\text{max}p}} x_{i,j}$ and find all ranges of $\theta$ for which $|\text{PartAvg}_{\theta_k}(y_p, t) - \text{PartAvg}_{\theta_k}(y_p, t+1)| > \beta$, where $\beta$ is a fixed threshold, and update the motion vector for each $y_p$.
6. Update the segment locations based on the calculated magnitude and direction of significant motion vectors and form a mask covering the dynamic area.
7. Calculate the number of samples required for reconstruction using Eq. (4).
8. Form a measurement matrix as $A = \text{Mask} \times \text{rand}(m, n)$ and use the measurements for reconstruction.
9. Go to step 3 if the area under the mask is less than a predefined dynamic area. Start from Step 2 if area is greater than the dynamic area using same number of clusters estimated using first frame and same threshold value in step 3.
```

$$\text{PartAvg}_k(y_p, t) = \frac{1}{s_p}\sum h_k \cdot f_t \quad \text{for } k = 1 \ldots 8, \tag{15}$$

where f_t is the frame at time t and h_k is a mask defined by

$$h_k = (c_p \cup y_p) \cap s_k.$$

Here, c_p is the circle around the centroid of segment y_p:

$$c_p(x, y) = \begin{cases} 1 & \sqrt{x^2 + y^2} < \text{rad}_p \\ 0 & \text{o.w} \end{cases} \tag{16}$$

and s_k is the k'th sector centered at rad_p:

$$s_k(x, y) = \begin{cases} 1 & \frac{\pi(k-1)}{4} < \theta_k < \frac{\pi k}{4} \\ 0 & \text{o.w} \end{cases}. \tag{17}$$

We find all θ_k, for which

$$|\text{PartAvg}_k(y_p, t) - \text{PartAvg}_k(y_p, t+1)| > \beta. \tag{18}$$

Figure 5 graphically represent the motion estimation procedure. In order to calculate the magnitude of change, we move the segment in the estimated direction θ_k for all possible x and y inside the circle and measure the average outside the segment for each coordinate pair. For pixel locations y_p,

$$y_p(x_i, y_i) = y_p(x + r_i \cos\theta_k, y + r_i \sin\theta_k),$$

where r_i is such that the distance of the farthest pixel in segment y_p remains less than the radius of the circle; i.e., $d(y_p, \text{cen}_p) < \text{rad}_p$. Using Eq. (15), we calculate all averages outside the segment boundary, maximizing the difference between the average calculated and the previous frame average over all pairs of coordinates. We then calculate the magnitude of the displacement m_{θ_k} in the θ_k direction:

$$\begin{aligned} m_{\theta_k} = max|&\text{PartAvg}_{\theta_k}[y_p(x, y), t+1] \\ &- \text{PartAvg}_{\theta_k}[y_p(x_i, y_i), t]|i \quad \forall\, i. \end{aligned} \tag{19}$$

The new coordinates for y_p in direction θ_k will be defined as

$$x' = x + m_{\theta_k}\cos\theta_k \qquad y' = y + m_{\theta_k} y \sin\theta_k.$$

We calculate the updated segment dynamic area by combining the translated segment in all significant θ_k directions. The updated segment dynamic area is found as

$$\begin{aligned} y_p(x, y) = y_p(x, y) &\cup y_p(x + m_{\theta_1}\cos\theta_1, y + m_{\theta_1}\sin\theta_1) \\ &\times \cup \ldots \cup y_p(x + m_{\theta_k}\cos\theta_k, y + m_{\theta_k}\sin\theta_k). \end{aligned} \tag{20}$$

Figure 6 graphically represent the magnitude estimation process. A mask for all dynamic segment areas can be formally expressed as

$$M_t = y_1 \cup y_2 \cup \ldots \cup y_p. \tag{21}$$

The measurement matrix then can be written as a scalar product of a Gaussian rand matrix and the mask:

$$A = \text{rand}(m, n) \times M_t. \tag{22}$$

A binary mask is created using the location information of the dynamic segments. By rewriting Eq. (4), the number of measurements $M3$ are assigned based on

$$M3 = \log(n)/[\varepsilon/(C \cdot S)]^{2p/(p-2)}, \tag{23}$$

where n is the sum of ones in the mask, S is the sparsity index of the last segmented frame dynamic area inside the mask, ϵ is the reconstruction error bound, C is a constant value dependent on the correlation between the sampling and basis matrix, and p is taken as $2/3$. The number of measurements k is bounded below by $0.3n$ and bounded above by $0.6n$.

In order to form a measurement matrix, we need to transmit the information about segments corresponding to each pixel. The location of each segment is shared with the encoder after resegmentation is performed at the decoder end. Therefore, the bandwidth required per frame depends on the resegmentation interval and estimated number of segments required using contrast and variance information from Eqs. (12) and (13). We calculate the measurements required to transmit the information using the relationship shown here:

$$M1 = \frac{\text{Total Number of pixels}}{(\text{bit depth} - \text{bits req to represent segments})\text{Resegmentaion interval}}, \tag{24}$$

where $M1$ is the number of measurements per frame used to transmit the mask. The number of measurements required for motion estimation is calculated using the following equation:

$$M2 = \text{direction resolution} \times \text{Avg.dynamic segments/frame} + \text{No. of segments}, \tag{25}$$

where $M2$ is the number of measurements per frame required to estimate the motion of dynamic segments. These measurements are used at the encoder to generate a new mask and transmitted from encoder to decoder for updating the mask at the decoder for reconstruction. The total number of measurements can be expressed as

$$M = M1 + M2 + M3.$$

After acquisition of the measurements from the dynamic area, we use the total variation minimization algorithm for estimation in a manner similar to Eq. (8):

$$\text{minimize}\|x\|_{\text{TV}} \quad \text{subject to } y = Ax. \tag{26}$$

All the dynamic pixels are replaced by new estimates and static pixel values are taken from the previous frame:

$$f_{t+1} = M_t^c \cdot f_t + M_t \cdot x,$$

where M_t^c is the complement of M_t.

When the area under the mask is less than a predefined percentage of the whole frame κ averages for each segment are measured and any segment other than the previously selected segment with a changed average above the threshold is included in the mask. Above-threshold segments are processed further for motion detection and mask creation for the next frame. Re-segmentation is performed when the κ value jumps above a predefined percentage.

We have found this scheme efficient for surveillance videos with complex background and slow changes. Masking the static background using segmentation reduces the number of pixels to be estimated with greater precision, thereby increasing the performance of the reconstruction algorithm.

4 Simulations and Analysis

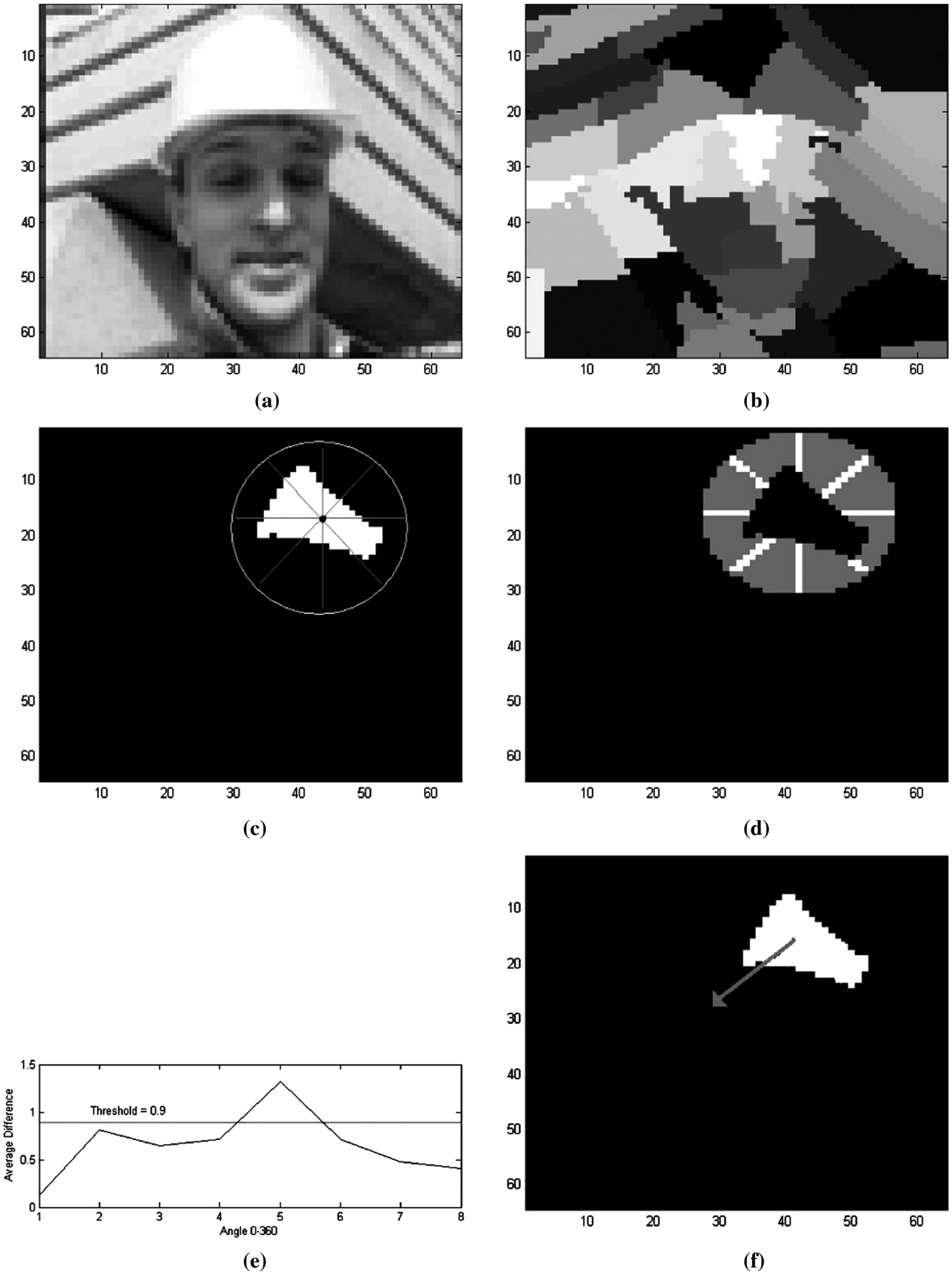
In the experimental studies, we first show the variation of subrate and PSNR for the proposed technique. Simulated and real videos were obtained for this study. Each video was created or downsampled to a size of 64×64 pixels per frame. The machine used for simulation has a 2.4-GHz processor and 4-GB RAM. The simulated videos are of a human-shaped object moving linearly across a uniform background at different speeds. The real videos are taken with a thermal infrared camera and show animals moving at different speeds under the control of human handlers. These videos are representative of the types of scenes expected to be encountered in a practical implementation of our algorithm in a sensor. The quantization is assumed to be 16 bits for calculation of $M1$ and $M2$. The subrate is controlled by varying the multiplicative constant C_m from 0.1 to 0.8 in the following expression $M3 = C_m \times n$, where n is the sum of ones in the mask. The κ threshold was set at 0.1 for simulated video and 0.5 for real video sequences. Here, values of κ are chosen according to the changes in video. This also show the effect of κ on the number of motion estimation (ME) and mask measurements. All measurements are averaged over 30 frames. The results are shown in Table 1 for a simulated video and real video and plotted in Fig. 7. The ME measurements for simulated video are less due to the complexity of scene. Mask measurements are high due to a smaller percentage of area threshold. In real video, the mask measurements are less due to a higher percentage of area threshold and ME measurements are high due to a greater number of segments selected to separate the foreground and background.

In order to demonstrate the performance of the proposed technique, we performed a simulation using simulated and real video sequences and recorded the PSNR and the subrate used for each frame. In simulated videos, temporal changes are varied for eight video sequences over a minimum to a maximum range of temporal changes. The texture in each frame is kept to a minimum in order to minimize the number of segments required for separating foreground and background. A random two-dimensional signal with 10^{-4} variance is added to each frame as well to simulate small changes. Change is calculated based on the following expression:

$$\Delta = \text{MSE}[f(t-1), f(t)]. \tag{27}$$

The subrate and PSNR is recorded for each reconstructed video sequence and compared with intraframe TV, frame differencing, and BCS-SPL-CT methods. Real videos were recorded using a longwave infrared camera in a fixed position in a natural environment. The video is of an animal passing through the field of view at different speeds controlled by a human.

The proposed method reduces the computational complexity of the reconstruction algorithm and produces a frame in less time than the other methods when the change is below a threshold. It adapts the subrate according to the changes in a video. An average subrate over 30 frames for a particular video using our proposed method is used for reconstruction using the other methods. The time requirements and PSNR for intraframe TV and BCS-SPL-CT are irrespective of the changes in the video but depend on the subrate. As mentioned before, the subrate of our adaptive method is passed to the other methods for reconstruction, which changes the PSNR and time accordingly. Therefore, we have used a ratio of PSNR to seconds per frame in order to demonstrate the performance comparison. As shown in Fig. 8, the ratio is the maximum for least change



(a)

(b)

(c)

(d)

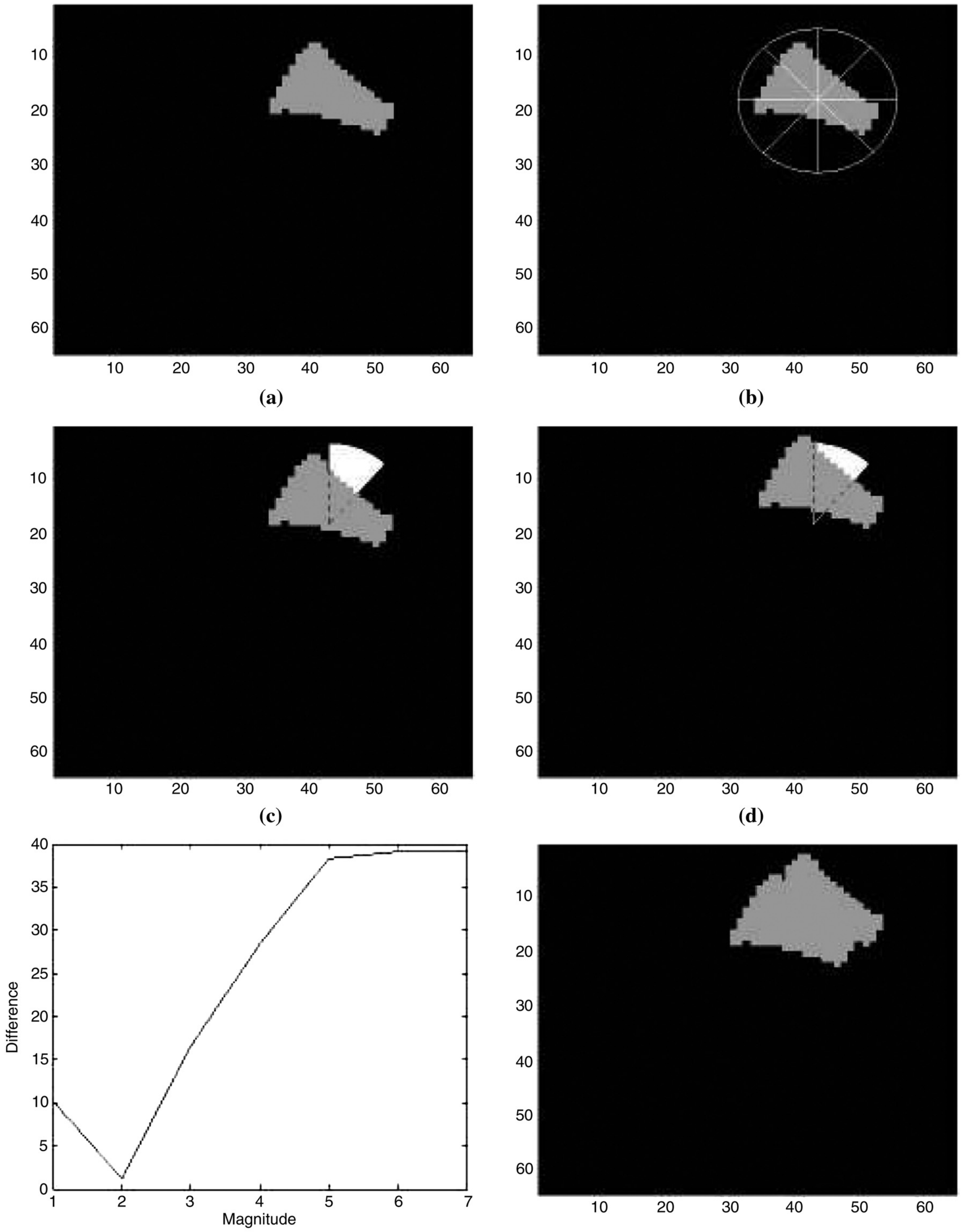
(e)

(f)

Fig. 5 (a) Foreman frame. (b) Segmented foreman frame. (c) One of the changed segment partitioned into eight parts. (d) Area outside the segment partitioned into eight parts.(e) $\text{PartAvg}_{\theta_k}(y_f, t) - \text{PartAvg}_{\theta_k}(y_f, t+1)$ for same x. (f) Direction of movement.

for our proposed method and the differencing algorithm and drops as the change increases.

The PSNR is also compared to the existing methods and it is noted that results using the proposed method holds the PSNR value in the 2-db range, while other algorithms PSNR drops as the temporal changes are decreased. The reason for the drop is the use of an adaptive subrate required by the proposed method which adapts to less changes while the other algorithms, with the exception of differencing, reconstruct irrespective of changes in a scene. The change in subrate required by our proposed technique is not pronounced in the simulated video as compared to real videos.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Fig. 6 (a) A dynamic segment of Foreman video. (b) Initial segment position, $\text{PartAvg}_{\theta=45°}(y_f, t)$. (c) Segment position incremented in a single direction and $\text{PartAvg}_{\theta=45°}(y_f, t)$ for x_{i_1,j_1}. (d) Another segment position $\text{PartAvg}_{\theta=45°}(y_f, t)$ for x_{i_2,j_2}. (e) $\text{PartAvg}_{\theta_k}(y_f, t) - \text{PartAvg}_{\theta_k}(y_f, t+1)$ for all θ. (f) Final dynamic area for measurements.

This affects the performance of the other algorithms curves in both cases. The differencing algorithm shows a less steep downward trend compared to our proposed algorithm.

Figure 9 shows five frames from four original and reconstructed simulated videos with D increasing from the top down, following the increased speed of the object. Figure 10 shows five frames from four original and reconstructed real videos, recorded at a trail using a long-wave infrared camera. The κ threshold was kept at 0.3 and 0.5 for simulated and real video, respectively, for reconstruction. The parameter C as empirically chosen to be 1.5 and ϵ was taken to be 0.1. The parameter S was calculated based on a threshold taken as 0.13 in Eq. (23). The first real surveillance video was reconstructed with about 6%

Table 1 The breakup of subrate into measurement of mask transmission (M1), ME (M2), and scene measurements (M3) for a simulated video sequence and a real video sequence.

Sr. No.	*M*1	*M*2	*M*3	*M*	Subrate	PSNR
			Simulated video sequence			
1	82	10	149	241	0.05	27.8
2	82	10	176	268	0.06	28.9
3	82	10	201	293	0.07	35.5
4	82	10	247	339	0.08	42.7
5	82	10	327	419	0.102	44.5
6	82	10	340	432	0.105	49
7	82	10	355	447	0.109	50
8	82	10	399	491	0.119	52.5
			Real video sequence			
1	63	60	162	285	0.069	27.2
2	63	60	272	395	0.096	29.5
3	63	60	331	454	0.110	32.7
4	63	60	366	489	0.119	35.2
5	63	60	571	694	0.169	37.2
6	63	60	574	697	0.169	38.4
7	63	60	768	891	0.217	39.9
8	63	60	827	950	0.231	40

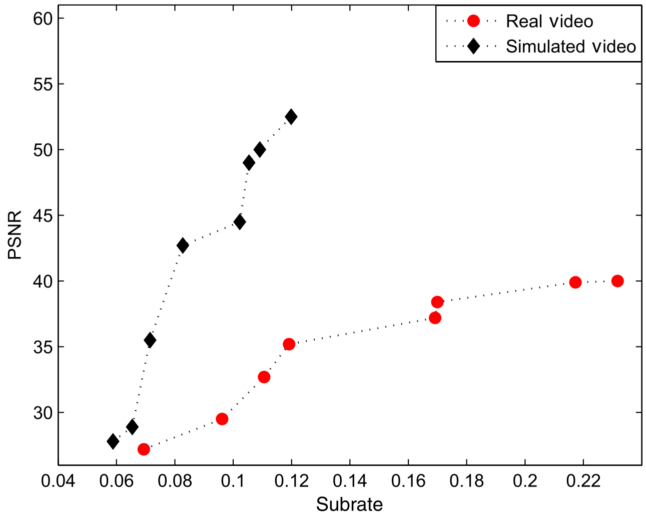


Fig. 7 The PSNR-to-subrate curve for simulated video sequence and real video sequence.

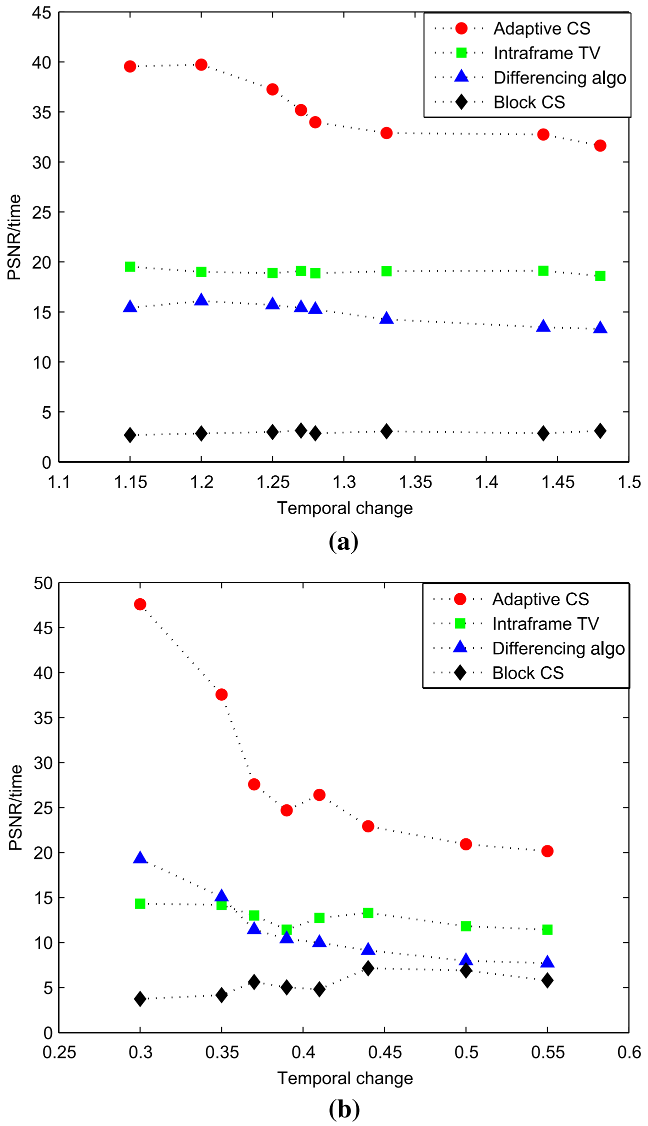


Fig. 8 PSNR/time versus temporal change curves for (a) simulated video sequence; (b) real video sequence.

of the measurements necessary for each frame on average, compared to traditional raster scanning. The number of samples falls to only 1% for a few frames where most of the dynamic area over the whole frame is below threshold. These measurements are used to reconstruct only the dynamic area, which is on average about 25% per frame for this video. This takes the computational load from the optimization algorithm, hence improving the time required for reconstruction. This number can be further improved if we assign some parameters, such as a threshold and number of clusters, individually to each sequence based on the characteristics of the sequence. Basically the scheme tracks an object once it is detected in motion. Prediction of the direction and magnitude of motion enables us to assign measurements strategically and improve reconstruction efficiency. Due to the shape and texture of segments, some static areas are picked up and some dynamic areas fall below threshold. The algorithm checks for change before collecting measurements and incorporates the new dynamic areas in the next frame so that there is minimal residual error

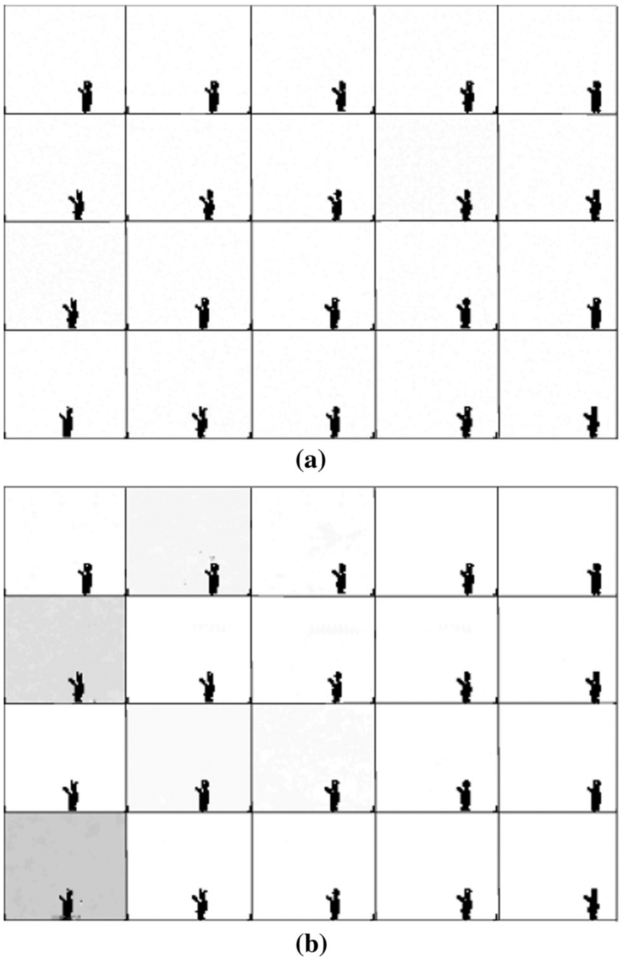


Fig. 9 (a) Simulated motion of person across a frame at different speeds increasing from top to bottom. (b) Reconstructed frames of simulated motion in (a).

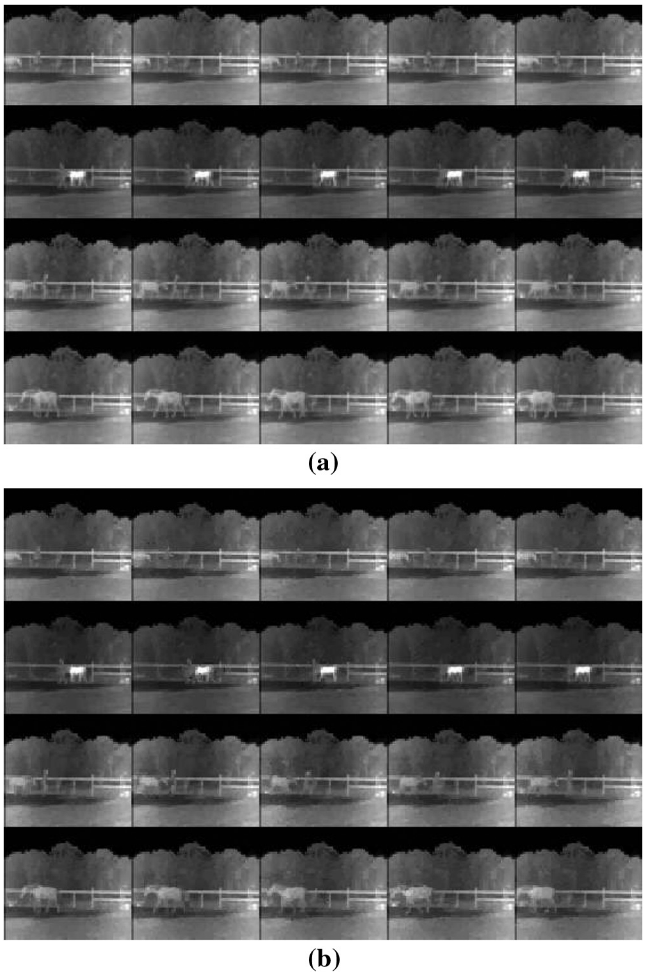


Fig. 10 (a) Real videos of animals walking across a frame with increasing MSE top to bottom. (b) Reconstructed videos frames in (a).

accumulation. The residual error may accumulate in areas classified as static. In all test cases in this study, the errors were removed within a few frames.

Our interest in CS techniques is in applying them to problems related to surveillance videos. Most researchers applying CS to video are interested in a more general application of this technique to all types of video. The videos we used represent the types of videos that we expect to encounter in our applications. However, our restriction to these types of videos leaves open the question of how our technique would work against more traditional video sequences. To address this question, we show the results for the Foreman video sequence in Table 2. This video was downsampled to the same 64×64 pixel size as used for the other videos. This video is very different in character from the ones shown previously in this article. The amount of change between frames as indicated by the D parameter is an order of magnitude bigger. As with most of the other videos, our technique has PSNR values second only to the frame differencing method. However, the Block CS technique in this case outperforms our method significantly with respect to execution time. This is to be expected since our algorithm scales with the number and size of segments changing, and there is a significant amount of change in this video. The Block CS method uses Contourlets as a sparsifying transform and seems to be more effective when the changes are larger. Since reconstruction in Block CS is done in the sparse domain and then transformed, this indicates that the Foreman video possess greater sparsity in this domain than the other videos used in this study. While this indicates that our algorithm loses performance for the conditions inherent in the Foreman video, the prior results presented in this article indicate that Block CS loses performance when the change in videos is small. A full exploration of the reasons for this behavior is left for future study.

In order to see effects due to segmentation, the number of segments was varied keeping the parameter κ constant. PSNR was recorded against total subrate, which is the sum of scene measurements, ME measurements, and mask transmission measurements and is plotted in Fig. 11. The numbers above the points in Fig. 11 represent the number of segments used. The PSNR does trend upward with an increase in subrate, but is not strictly monotonic function of subrate. There is very little apparent relationship between the number of segments and PSNR. ME measurements and mask transmission measurements are directly related to the number of segments for simulated and real

Table 2 Foreman video sequence comparisons with three other methods.

							Adaptive CS			Intraframe CS			Frame differencing			Block CS		
Video	**Δ**	***M*1**	***M*2**	***M*3**	***M***	**Subrate**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**
Foreman	3.4	136	168	989	1294	0.32	27.2	3.4	7.8	25.4	3.3	7.6	33.5	12.1	2.7	23.5	0.9	25.1

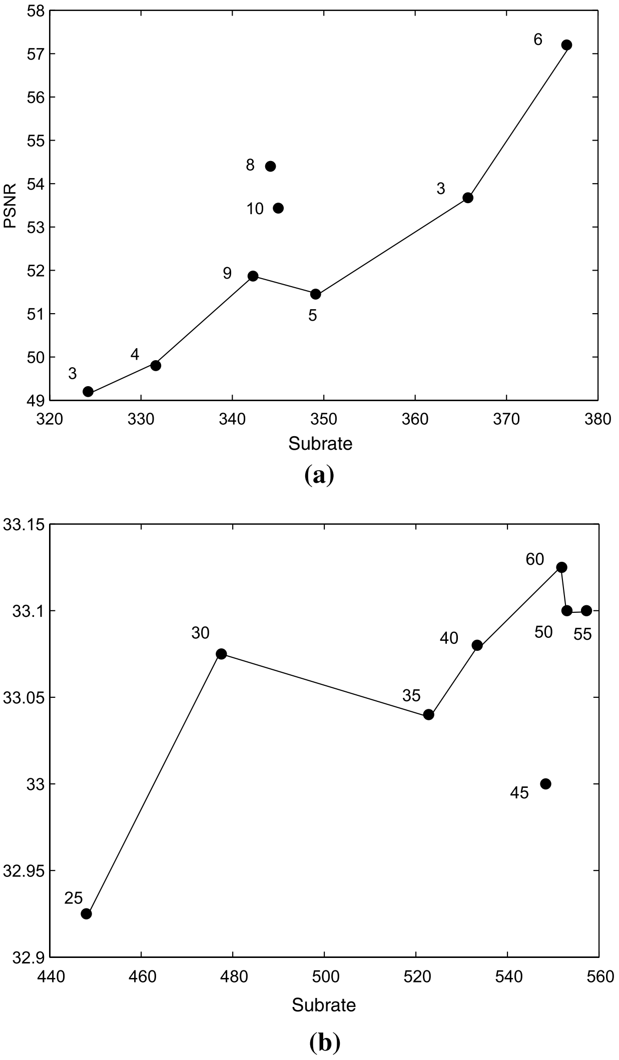
videos but they do not appear to directly relate to PSNR. Each video has an optimal number of segments for which PSNR is maximum. We believe that PSNR is primarily reflecting the characteristics of the segmentation algorithm used. As a result, further analysis was deemed outside the original scope of this paper and will be pursued in future research.

After mask transmission and until resegmentation, measurement allocation in our algorithm is performed at the sensor level with less complexity and higher speed than previous methods. The method is adaptive to the complexity of the scene. A change in average from a previous frame is calculated before sampling. At the decoder, only dynamic pixels are reconstructed reducing the complexity by the number of static pixels. Some methods based on motion estimation and motion compensation, such as ME/MC BCS-SPL and MH-BCS-SPL, accumulate the measurements for a series of frames and perform reconstruction of all frames simultaneously.[6–8] The method proposed here reduces the complexity of the optimization process used in reconstruction not only by single frame reconstruction, but also by reconstructing only the dynamic area of each frame. Scenes where the dynamic area is small and the motion is not complex can be potentially reconstructed in real time. The resegmentation interval is adaptive to the complexity and the spread of motion in a video as shown in Tables 3 and 4, thereby reducing the computational requirements for segmentation. In addition, if the slow change assumption holds, segmentation of previous frames can be used for forming the mask and performing motion estimation, which removes the constraint of generating new masks after reconstruction.

In a feature comparison to existing adaptive block-based techniques, which require optimization of measurement allocation before each frame on the decoder side, this technique can acquire a number of frames with far less computational time required on the encoder side depending on the dynamic areas spread in the scene.[11] We have observed that if the segmentation is of good quality then this method is efficient for surveillance videos in terms of computations and sampling efficiency.

In this study, we have taken the last reconstructed frame for resegmentation but in order to avoid latency due to the segmentation process, we can also make it a parallel background process. Segmentation can be performed after each reconstruction, and the last available segmented frame before the mask area upper bound is reached can be used to avoid any delays in making measurements. This scheme basically reduces computations by load sharing between two routines. It can be further improved by implementing better parameter estimation techniques and by optimizing the process of measuring averages for motion detection. The TV minimization package used for reconstruction in this paper had the parameters shown in Table 5.[29]

In order to give an overview of computational time requirements to realize this technique, we calculated some values based on the first real video results. The nominal flipping rates of 200 ns for a micromirror array are taken from the advertised specifications of the device.[30] The sampling and transmission is assumed to be performed at the same rate. The total time for measurement acquisition, motion estimation, mask update, and transmission latency is calculated



(a)

(b)

Fig. 11 PSNR variation with subrate by changing number of segments. (a) Simulated video. (b) Real video, the number beside each data point shows number of segments used.

Table 3 Simulated video sequence comparisons with three other methods.

							Adaptive CS			Intraframe CS			Frame differencing			Block CS		
	Δ	***M*1**	***M*2**	***M*3**	***M***	**Subrate**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**
1	1.15	36	38	199	273	0.066	37.7	0.95	39.5	35.8	1.83	19.5	45.7	2.9	15.4	14.1	5.2	2.67
2	1.2	36	38	205	279	0.067	38.4	0.96	39.7	36.1	1.9	19	45.6	2.83	16.04	14.19	5	2.83
3	1.25	36	38	215	288	0.073	38.5	1.03	37.2	36.5	1.93	18.8	45.5	2.9	15.6	15.8	5.3	2.98
4	1.27	36	50	220	306	0.074	38.7	1.1	35.1	36.9	1.93	19.0	45.7	2.96	15.4	15.7	5	3.14
5	1.28	36	50	221	307	0.074	38.5	1.13	33.9	37.1	1.96	18.8	45.7	3	15.2	15.78	5.5	2.85
6	1.33	36	50	225	311	0.075	38.8	1.18	32.8	37.5	1.96	19.0	45.6	3.2	14.25	15.7	5.13	3.05
7	1.44	36	50	230	316	0.077	39.5	1.2	32.7	37.6	1.96	19.11	45.3	3.3	13.45	15.12	5.3	2.85
8	1.48	36	50	235	321	0.078	39	1.23	31.6	37.8	2.03	18.5	45.2	3.4	13.29	15.6	5.03	3.09

Table 4 Real video sequence comparisons with three other methods.

							Adaptive CS			Intraframe TV			Frame differencing			Block CS		
	Δ	***M*1**	***M*2**	***M*3**	***M***	**Subrate**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**	**PSNR**	***t* s**	**Ratio**
1	0.3	49	50	179	278	0.06	34.9	0.73	47.5	23.4	1.63	14.3	46.3	2.4	19.2	17.3	4.6	3.7
2	0.35	49	50	221	320	0.07	33.8	0.9	37.5	24.6	1.73	14.19	45.7	3.03	15.0	16.9	4.06	4.15
3	0.37	49	50	324	423	0.10	33.1	1.2	27.5	26	2	13	45.7	4	11.4	21.2	3.7	5.6
4	0.39	49	63	346	458	0.113	32.1	1.3	24.6	26.3	2.3	11.4	44.8	4.3	10.4	19.1	3.8	5.02
5	0.41	49	63	361	473	0.115	36.1	1.36	26.4	25.5	2	12.7	44.3	4.43	9.99	18.3	3.8	4.8
6	0.44	49	63	376	488	0.118	32.1	1.4	22.9	27.5	2.06	13.3	45.02	4.93	9.1	22.4	3.1	7.14
7	0.5	49	63	373	485	0.118	31.4	1.5	20.9	26.8	2.2	11.8	40.7	5.1	7.9	21.2	3.0	6.91
8	0.55	49	63	408	520	0.12	33.6	1.6	20.16	26.3	2.3	11.4	41.7	5.4	7.7	19.7	3.4	5.79

Table 5 Parameters used for TV minimization.

Parameter	Value
opt.mu	2^8
opt.beta	2^5
opt.tol	1×10^{-3}
Opt.maxit	300
opt.TVnorm	1
opt.nonneg	true

Table 6 Computational time bounds for real-time reconstruction.

Task	Analytical time	Time requirement (ms)
Measurement acquisition	$M3 \times$ sampling rate	0.09
Motion estimation	$M2 \times$ sampling rate$+$ subraction time	0.0009
Mask update	transmission rate $\times$ $M1$	0.0002
Transmission latency	$(M1 + M2 + M3)\times$ sampling rate	0.10
Reconstruction ∥ Segmentation	0.33 – Total time	320
All		<0.33 s

and subtracted from the minimum time to render a frame for real-time streaming. This time bound can be used to complete reconstruction and segmentation in parallel. The results are listed in Table 6.

To summarize the important points: in the proposed technique, after mask transmission and until resegmentation, the measurement allocation is performed at the sensor level with less complexity and high speed. The method is adaptive to the complexity of the scene. A change in average from the previous frame is calculated before sampling. At the decoder, only dynamic pixels are reconstructed, reducing the complexity and thereby increasing the efficiency of reconstruction algorithms.

5 Conclusions and Future Work

We have discussed in this article a new scheme to acquire measurements for video reconstruction. We found this scheme useful for temporal compression in videos with static background and slow foreground changes over time. Depending on the video, this scheme is able to decrease reconstruction time and computations compared to some existing sampling techniques. Furthermore, the motion estimation is very efficient from a hardware implementation perspective. However, while we believe that we have made a good case that the computational burden of this algorithm is actually less than most other CS techniques for video, it should not be compared to traditional video compression. Given the fact that memory is inexpensive, visible band cameras are inexpensive, and hardware coding/decoding is fairly inexpensive for traditional video devices, the desirability of any CS technique is limited for visible band sensors. However, our scheme can prove useful at wavelengths where an array of sensors is expensive, and single-pixel detection is the most cost-efficient method for producing video. Examples would include terahertz sensors[31] and perhaps infrared.[32] The gains from this scheme can be applied to reducing reconstruction time and computational requirements or increasing frame rates for video imaging at wavelengths where sensor arrays are expensive. In order to improve the results shown here, use of spatial sparsity transform domain knowledge incorporated with the sampling matrix could prove fruitful. Further studies of the impact of parameters such as the radius of the circle and shape to encircle should also be performed. The motion estimation scheme can be optimized to use least averages for predicting the direction. Further, other robust methods can be investigated for parameter estimation. Noise should be taken into consideration to study the effects on performance and a denoising scheme designed for and applied to reducing the artifacts due to quantization and detection noise. We are constructing a hardware simulator of a single-pixel camera device for testing and further development of this algorithm.

Acknowledgments

The authors wish to thank the anonymous reviewers for many helpful comments during the preparation of this paper. The authors also acknowledge the aid of our colleagues, Drs. Orges Furxhi and Srikant Chari, in obtaining and preparing the simulated and real videos used in this study.

References

1. R. G. Baraniuk et al., "A new compressive imaging camera architecture using optical-domain compression," *Proc. SPIE* **6065**, 606509 (2006).
2. R. Robucci et al., "Compressive sensing on a CMOS separable transform image sensor," *Proc. IEEE* **98**(6), 1089–1101 (2010).
3. R. F. Marcia and R. M. Willett, "Compressive coded aperture video reconstruction," in *Proc. European Signal Process. Conf.*, Vol. 1, pp. 1–5 (2008).
4. J. Zheng and E. L. Jacobs, "Video compressive sensing using spatial domain sparsity," *Opt. Eng.* **48**(8), 087006 (2009).
5. A. C. Sankaranarayanan et al., "Compressive acquisition of dynamical scenes," in *Proc. 11th European Conf. on Comput. Vis.*, pp. 129–142, Springer-Verlag, Berlin, Heidelberg (2010).
6. J. E. Fowler, S. Mun, and W. Tramel, "Block-based compressed sensing of images and video," *Found. Trends Sig. Process.* **4**(4), 297–416 (2012).
7. E. W. Tramel and J. E. Fowler, "Video compressed sensing with multihypothesis," in *Proc. IEEE Data Compression Conf.*, pp. 193–202, IEEE, Snowbird, Utah (2011).
8. H. Jung et al., "k-t FOCUSS: a general compressed sensing framework for high resolution dynamic MRI," *Mag. Res. Med.* **61**(1), 103–116 (2009).
9. J. E. Fowler, S. Mun, and E. W. Tramel, "Multiscale block compressed sensing with smoother projected Landweber reconstruction," in *Proc. European Signal Process. Conf.*, Barcelona, Spain, pp. 564–568 (2011).
10. S. Mun and J. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Proc. IEEE Data Compression Conf.*, pp. 183–192, IEEE, Snowbird, Utah (2011).
11. Zhaorui Liu, A. Y. Elezzabi, and H. Vicky Zhao, "Maximum frame rate video acquisition using adaptive compressed sensing," *IEEE Trans. Circ. Syst. Video Technol.* **21**(11), 1704–1718 (2011).
12. H. Nyquist, "Certain topics in telegraph transmission theory," *Proc. IEEE* **90**(2), 280–305 (2002).
13. E. J. Candès and Y. Plan, "A probabilistic and RIPless theory of compressed sensing," *IEEE Trans. Inform. Theor.* **57**(11), 7235–7254 (2011).
14. R. Baraniuk et al., "A simple proof of the restricted isometry property for random matrices," *Construct. Approx.* **28**(3), 253–263 (2008).
15. L. Gan et al., "Analysis of the statistical restricted isometry property for deterministic sensing matrices using Steins method," Preprint (2009).
16. D. L. Donoho and J. Tanner, "Neighborliness of randomly projected simplices in high dimensions," *Proc. Natl. Acad. Sci. U. S. A.* **102**(27), 9452–9457 (2005).
17. J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theor.*, **53**(12), 4655–4666 (2007).
18. S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.* **57**(7), 2479–2493 (2009).
19. Z. Zhang and B. D. Rao, "Iterative reweighted algorithms for sparse signal recovery with temporally correlated source vectors," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, pp. 3932–3935, IEEE, Prague, Czech Republic (2011).
20. P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Trans. Image Process.* **7**(3), 304–309 (1998).
21. E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique* **346**(9–10), 589–592 (2008).
22. W. Xu, "Compressive sensing for sparse approximations: constructions, algorithms and analysis," Master's Thesis, California Institute of Technology (2010).
23. E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *IEEE Trans. Inform. Theor.* **52**(12), 5406–5425 (2006).
24. J. N. Laska et al., "Democracy in action: Quantization, saturation, and compressive sensing," *Appl. Comp. Harm. Anal.* **31**(3), 429–443 (2011).
25. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 731 (1997).
26. P. L. Correia and F. Pereira, "Objective evaluation of video segmentation quality," *IEEE Trans. Image Process.* **12**(2), 186–200 (2003).
27. P. L. Rosin, "Unimodal thresholding," *Pattern Recognit.* **34**(11), 2083–2096 (2001).
28. I. Noor and E. L. Jacobs, "Adaptive compressive sensing algorithm for video acquisition using a single-pixel camera," *Proc. SPIE* **8365**, 83650J (2012).
29. Li Chengbo, Yin Wotao, and Zhang Yin, "TV Minmization," http://www.caam.rice.edu/optimization/L1/TVAL3 (2010).
30. T. Instruments, "DLP5500 Digital Micromirror Device," datasheet (2012).
31. W. L. Chan et al., "A single-pixel terahertz imaging system based on compressed sensing," *Appl. Phys. Lett.* **93**(12), 12110–12115 (2008).
32. R. M. Willett, R. F. Marcia, and J. M. Nichols, "Compressed sensing for practical optical imaging systems: a tutorial," *Opt. Eng.* **50**(7), 072601 (2011).

Imama Noor is a PhD student in the Department of Electrical and Computer Engineering at the University of Memphis in Tennessee. She is currently working as a staff engineer at the Arcon Corporation, Waltham, Massachusetts. Her research interests include applications of image sensing and image processing. She received MS degrees in electronics and computer engineering from Quaid-i-Azam University, Islamabad, Pakistan, and University of Engineering and Technology Taxila, Pakistan, respectively.

Eddie L. Jacobs is an associate professor in the Department of Electrical and Computer Engineering at the University of Memphis. His research interests are in novel imaging sensor development, electromagnetic propagation and scattering, and human performance modeling. He received BS and MS degrees in electrical engineering from the University of Arkansas and a DSc in electrophysics from George Washington University.