Reconstruction-free action inference from compressive imagers

Kuldeep Kulkarni, Pavan Turaga

Abstract—Persistent surveillance from camera networks, such as at parking lots, UAVs, etc., often results in large amounts of video data, resulting in significant challenges for inference in terms of storage, communication and computation. Compressive cameras have emerged as a potential solution to deal with the data deluge issues in such applications. However, inference tasks such as action recognition require high quality features which implies reconstructing the original video data. Much work in compressive sensing (CS) theory is geared towards solving the reconstruction problem, where state-of-the-art methods are computationally intensive and provide low-quality results at high compression rates. Thus, reconstruction-free methods for inference are much desired. In this paper, we propose reconstruction-free methods for action recognition from compressive cameras at high compression ratios of 100 and above. Recognizing actions directly from CS measurements requires features which are mostly nonlinear and thus not easily applicable. This leads us to search for such properties that are preserved in compressive measurements. To this end, we propose the use of spatio-temporal smashed filters, which are compressive domain versions of pixel-domain matched filters. We conduct experiments on publicly available databases and show that one can obtain recognition rates that are comparable to the oracle method in uncompressed setup, even for high compression ratios.

Index Terms—Compressive Sensing, Reconstruction-free, Action recognition

1 Introduction

Action recognition is one of the long standing research areas in computer vision with widespread applications in video surveillance, unmanned aerial vehicles (UAVs), and real-time monitoring of patients. All these applications are heavily resource-constrained and require low communication overheads in order to achieve real-time implementation. Consider the application of UAVs which provide real-time video and high resolution aerial images on demand. In these scenarios, it is typical to collect an enormous amount of data, followed by transmission of the same to a ground station using a low-bandwidth communication link. This results in expensive methods being employed for video capture, compression, and transmission implemented on the aircraft. The transmitted video is decompressed at a central station and then fed into a action recognition pipeline. Similarly, a video surveillance system which typically employs many high-definition cameras, gives rise to a prohibitively large amount of data, making it very challenging to store, transmit and extract meaningful information. Thus, there is a growing need to acquire as little data as possible and yet be able to perform high-level inference tasks like action recognition reliably.

Recent advances in the areas of compressive sensing (CS) [1] have led to the development of new sensors like compressive cameras (also called single-pixel cameras (SPCs)) [2], which enable the acquisition of **'more for less'** by greatly reducing the amount of sensed data while preserving most of its information. Another compelling application of SPC is in the area of infrared imaging. It is well known that short-wave infrared (SWIR) cameras have applications in military surveillance and maritime navigation because of their ability to 'see-through' in poor-light environmental conditions like fog, smoke, haze etc. However, the cost of a SWIR pixel is prohibitively expensive, and this has prevented infrared cameras from being employed in the applications outlined above. SPCs provide a cost-effective solution for image acquisition in such spectral regions. The SPC employs just a single photodiode sensitive to wavelengths of interest and a micro-mirror array to acquire images. This greatly reduces the cost of the camera. More recently, InView Technology Corporation applied CS theory to build commercially available CS workstations and SWIR cameras, thus equipping CS researchers with a hitherto unavailable armoury to conduct experiments on real CS imagery. ***The goal of this paper is to investigate the utility of compressive cameras for inference tasks in computer vision (specifically action recognition) in improving the tradeoffs between reliability of performance and computational/storage load of the system in a resource constrained setting.*** SPCs differ from the conventional cameras in that they integrate the process of acquisition and compression by acquiring a small number of linear projections of

- *K. Kulkarni and P. Turaga are with the School of Arts, Media and Engineering and School of Electrical, Computer and Energy Engineering, Arizona State University. Email: kkulkar1@asu.edu, pturaga@asu.edu.*

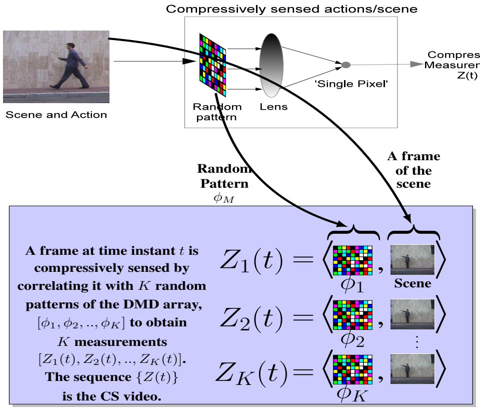

Compressively sensed actions/scene
Scene and Action
Random pattern
Lens
'Single Pixel'
Compressive Measurements Z(t)
Random Pattern ϕ_M
A frame of the scene
A frame at time instant t is compressively sensed by correlating it with K random patterns of the DMD array, $[\phi_1, \phi_2, .., \phi_K]$ to obtain K measurements $[Z_1(t), Z_2(t), .., Z_K(t)]$. The sequence $\{Z(t)\}$ is the CS video.
$Z_1(t) = \langle \phi_1, \text{Scene} \rangle$
$Z_2(t) = \langle \phi_2, \rangle$
$Z_K(t) = \langle \phi_K, \rangle$
Scene
⋮

Fig. 1. Compressive Sensing (CS) of a scene: Every frame of the scene is compressively sensed by optically correlating random patterns with the frame to obtain CS measurements. The temporal sequence of such CS measurements is the CS video.

the original images. More formally, when a sequence of images is acquired by a compressive camera, the measurements are generated by a sensing strategy which maps the space of $P \times Q$ images, $I \in \mathbb{R}^{PQ}$ to an observation space $Z \in \mathbb{R}^K$,

$$Z(t) = \phi I(t) + w(t), \tag{1}$$

where ϕ is a $K \times PQ$ measurement matrix, $w(t)$ is the noise, and $K \ll PQ$. The process is pictorially shown in Figure 1.

Difference between CS and video codecs: It is worth noting at this point that the manner in which compression is achieved by SPCs differs fundamentally from the manner in which compression is achieved in JPEG images or MPEG videos. In the case of JPEG, the images are fully sensed and then compressed by applying wavelet transform or DCT to the sensed data, and in the case of MPEG, a video after having been sensed fully is compressed using a motion compensation technique. However, in the case of SPCs, at the outset one does not have direct access to full blown images, $\{I(t)\}$. SPCs instead provide us with compressed measurements $\{Z(t)\}$ directly by optically calculating inner products of the images, $\{I(t)\}$, with a set of test functions given by the rows of the measurement matrix, ϕ, implemented using a programmable micro-mirror array [2]. While this helps avoid the storage of a large amount of data and expensive computations for compression, it often comes at the expense of employing high computational load at the central station to reconstruct the video data perfectly. Moreover, for perfect reconstruction of the images, given a sparsity level of s, state-of-the-art algorithms require $O(s \log(PQ/s))$ measurements [1], which still amounts to a large fraction of the original data dimensionality. Hence, using SPCs may not always provide advantage with respect to communication resources since compressive measurements and transform coding of data require comparable bandwidth [3].

While a great body of work has focused on the theory and algorithms for signal recovery, much less attention has been paid to the question of whether it is possible to perform high-level inference directly on CS measurements without reconstruction. This question is interesting due to the following reasons: a) very often we want to know some property of the scene rather than the entire scene itself, b) good quality reconstruction results are difficult to achieve at compression ratios of 100 and above, and c) the parameters to be input to the reconstruction algorithm such as signal sparsity, sparsifying basis are not known, and are chosen in an ad-hoc manner. ***In this paper, we consider the specific problem of action recognition in videos, and show that it is indeed possible to perform action recognition at extremely higher compression ratios, by bypassing reconstruction.*** We first show that approximate correlational features can be extracted directly from CS measurements. Using this in conjunction with the widely used correlational filters approach to recognition tasks in computer vision, we propose a spatio-temporal smashed filtering approach to action recognition, which results in robust performance at extremely high compression ratios.

1.1 Related work

a) Action Recognition: The approaches in human action recognition from cameras can be categorized based on the low level features. Most successful representations of human action are based on features like optical flow, point trajectories, background subtracted blobs and shape, filter responses, etc. The current state-of-the-art approaches [9], [10] to action recognition are based on dense trajectories, which are extracted using dense optical flow. The dense trajectories are encoded by complex, hand-crafted descriptors like histogram of oriented gradients (HOG) [11] , histogram of oriented optical flow (HOOF) [12], HOG3D [13], and motion boundary histograms (MBH) [9]. However, the extraction of the above features involves various non-linear operations. This makes it very difficult to extract such features from compressively sensed images. For a detailed survey of action recognition, the readers are referred to [14].

b) Action recognition in compressed domain: Though action recognition has a long history in computer vision, little exists in literature to recognize actions in the compressed domain. Yeo *et al.*[15] and Ozer *et al.*[16] explore compressed domain action recognition from MPEG videos by exploiting the spatiotemporal local structure, induced by the motion compensation technique used for compression. However, as stated above, the compression in CS cameras is achieved by randomly projecting the individual frames of the video onto a much lower dimensional space and hence does not easily allow leveraging motion information of the video. CS imagery acquires global measurements, thereby do not preserve any local information in their raw form, making action recognition much more difficult in comparison.

c) Reconstruction-free inference from CS videos: Sankaranarayanan *et al.*[17] attempted to model videos as a LDS (Linear Dynamical System) by recovering parameters directly from compressed measurements, but is sensitive to spatial and view transforms, making it more suitable for recognition of dynamic textures than action recognition. Thirumalai *et al.*[18] introduced a reconstruction-free framework to obtain optical flow based on correlation estimation between two compressively sensed images. However, the method does not work well at very low measurement rates. Calderbank *et al.*[19] theoretically showed that 'learning directly in compressed domain is possible', and that with high probability the linear kernel SVM classifier in the compressed domain can be as accurate as best linear threshold classifier in the data domain. Recently, Kulkarni and Turaga [20] proposed a novel method based on recurrence textures for action recognition from compressive cameras. However, the method is prone to produce very similar recurrence textures even for dissimilar actions for CS sequences and is more suited for feature sequences as in [21].

d) Correlation filters in computer vision: Even though, as stated above, the approaches based on dense trajectories extracted using optical flow information have yielded state-of-the-art results, it is difficult to extend such approaches while dealing with compressed measurements. Earlier approaches to action recognition were based on correlation filters, which were obtained directly from pixel data [23], [24], [25], [22], [26], [27]. The filters for different actions are correlated with the test video and the responses thus obtained are analyzed to recognize and locate the action in the test video. Davenport *et al.*[28] proposed a CS counterpart of the correlation filter based framework for target classification. Here, the trained filters are compressed first to obtain 'smashed filters', then the compressed measurements of the test examples are correlated with these smashed filters. Concisely, smashed filtering hinges on the fact that correlation between a reference signal and an input signal is nearly preserved even when they are projected onto a much lower-dimensional space. In this paper, we show that spatio-temporal smashed filters provide a natural solution to reconstruction-free action recognition from compressive cameras. Our framework (shown in Figure 2) for classification includes synthesizing Action MACH (Maximum Average Correlation Height) filters [22] offline and then correlating the compressed versions of the filters with compressed measurements of the test video, instead of correlating raw filters with full-blown video, as is the case in [22]. Action MACH involves synthesizing a single 3D spatiotemporal filter which captures information about a specific action from a set of training examples. MACH filters can become ineffective if there are viewpoint variations in the training examples. To effectively deal with this problem, we also propose a quasi view-invariant solution, which can be used even in uncompressed setup.

Contributions: 1) We propose a correlation-based framework for action recognition and localization directly from compressed measurements, thus avoiding the costly reconstruction process. 2) We provide principled ways to achieve quasi view-invariance in a spatio-temporal smashed filtering based action recognition setup. 3) We further show that a single MACH filter for a canonical view is sufficient to generate MACH filters for all affine transformed views of the canonical view.

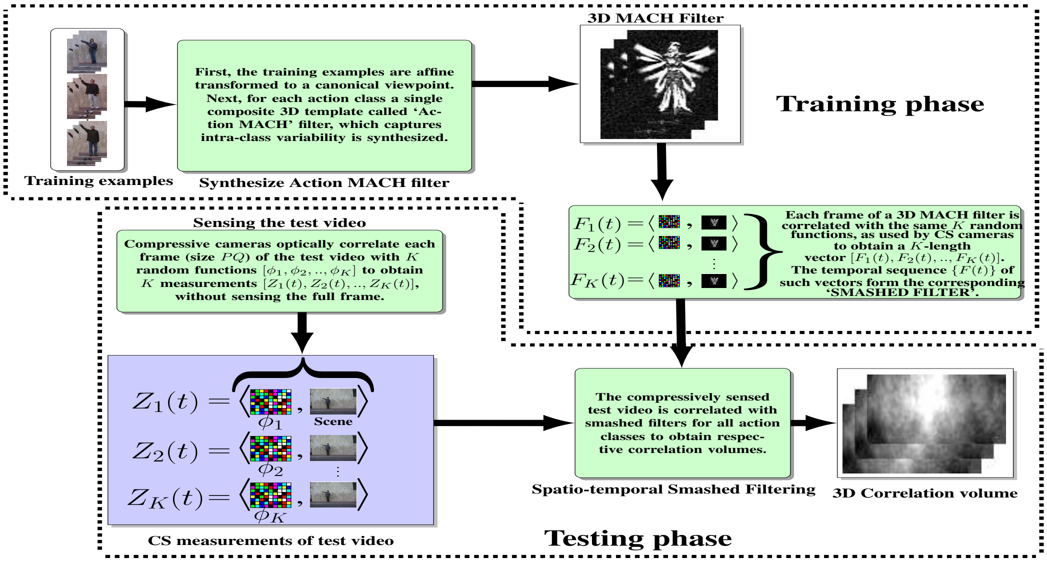
Outline: Section 2 outlines the reconstruction-free framework for action recognition, using spatio-temporal smashed filters (STSF). In section 3, we describe a quasi view-invariant solution to MACH based action recognition by outlining a simple method to generate MACH filters for any affine transformed view. In section 4, we present experimental results obtained on three popular action databases, Weizmann, UCF sports, UCF50 and HMDB51 databases.

2 Compressive action recognition

To devise a reconstruction-free method for action recognition from compressive cameras, we need to exploit such properties that are preserved robustly even in the compressed domain. One such property is the distance preserving property of the measurement matrix ϕ used for compressive sensing [1], [29]. Stated differently, the correlation between any two signals is nearly preserved even when the data is compressed to a much lower dimensional space. This makes correlation filters a natural choice to adopt. 2D correlation filters have been widely used in the areas of automatic target recognition and biometric applications like face recognition [30], palm print identification [31], etc., due to their ability to capture intraclass variabilities. Recently, Rodriguez *et al.*[22] extended this concept to 3D by using a class of correlation filters called MACH filters to recognize actions. As stated earlier, Davenport *et al.*[28] introduced the concept of smashed filters by implementing matched filters in the compressed domain. In the following section, we generalize this concept of smashed filtering to the space-time domain and show how 3D correlation filters can be implemented in the compressed domain for action recognition.

2.1 Spatio-temporal smashed filtering (STSF)

This section forms the core of our action recognition pipeline, wherein we outline a general method to im-


3D MACH Filter
Training phase
Training examples
Synthesize Action MACH filter
Sensing the test video
$F_1(t) = \langle \ , \ \rangle$
$F_2(t) = \langle \ , \ \rangle$
$F_K(t) = \langle \ , \ \rangle$
$Z_1(t) = \langle \ , \ \rangle$
ϕ_1
Scene
$Z_2(t) = \langle \ , \ \rangle$
ϕ_2
$Z_K(t) = \langle \ , \ \rangle$
ϕ_K
CS measurements of test video
Spatio-temporal Smashed Filtering
3D Correlation volume
Testing phase

First, the training examples are affine transformed to a canonical viewpoint. Next, for each action class a single composite 3D template called 'Action MACH' filter, which captures intra-class variability is synthesized.

Each frame of a 3D MACH filter is correlated with the same K random functions, as used by CS cameras to obtain a K-length vector $[F_1(t), F_2(t), .., F_K(t)]$. The temporal sequence $\{F(t)\}$ of such vectors form the corresponding 'SMASHED FILTER'.

Compressive cameras optically correlate each frame (size PQ) of the test video with K random functions $[\phi_1, \phi_2, .., \phi_K]$ to obtain K measurements $[Z_1(t), Z_2(t), .., Z_K(t)]$, without sensing the full frame.

The compressively sensed test video is correlated with smashed filters for all action classes to obtain respective correlation volumes.

Fig. 2. Overview of our approach to action recognition from a compressively sensed test video. First, MACH [22] filters for different actions are synthesized offline from training examples and then compressed to obtain smashed filters. Next, the CS measurements of the test video are correlated with these smashed filters to obtain correlation volumes which are analyzed to determine the action in the test video.

plement spatio-temporal correlation filters using compressed measurements without reconstruction and subsequently, recognize actions using the response volumes. To this end, consider a given video $s(x, y, t)$ of size $P \times Q \times R$ and let $H_i(x, y, t)$ be the optimal 3D matched filter for actions $i = 1, .., N_A$, with size $L \times M \times N$ and N_A is the number of actions. First, the test video is correlated with the matched filters of all actions $i = 1, ..N_A$ to obtain respective 3D response volumes as in (2).

$$c_i(l, m, n) = \sum_{t=0}^{N-1} \sum_{y=0}^{M-1} \sum_{x=0}^{L-1} s(l+x, m+y, n+t) H_i(x, y, t). \tag{2}$$

Next, zero-padding each frame in H_i upto a size $P \times Q$ and changing the indices, (2) can be rewritten as:

$$c_i(l, m, n) = \sum_{t=0}^{N-1} \sum_{\beta=0}^{Q-1} \sum_{\alpha=0}^{P-1} s(\alpha, \beta, n+t) H_i(\alpha-l, \beta-m, t). \tag{3}$$

This can be written as the summation of N correlations in the spatial domain as follows:

$$c_i(l, m, n) = \sum_{t=0}^{N-1} \langle S_{n+t}, H_i^{l,m,t} \rangle, \tag{4}$$

where, $\langle , \rangle$ denotes the dot product, S_{n+t} is the column vector obtained by concatenating the Q columns of the $(n+t)^{th}$ frame of the test video. To obtain $H_i^{l,m,t}$, we first shift the t^{th} frame of the zeropadded filter volume H_i by l and m units in x and y respectively to obtain an intermediate frame and then rearrange it to a column vector by concatenating its Q columns. Due to the distance preserving property of measurement matrix ϕ, the correlations are nearly preserved in the much lower dimensional compressed domain. To state the property more specifically, using JL Lemma [29], the following relation can be shown:

$$c_i(l, m, n) - N\epsilon \leq \sum_{t=0}^{N-1} \langle \phi S_{n+t}, \phi H_i^{l,m,t} \rangle \leq c_i(l, m, n) + N\epsilon. \tag{5}$$

The derivation of this relation and the precise form of ϵ is as follows. In the following, we derive the relation between the response volume from uncompressed data and response volume obtained using compressed data. According to JL Lemma [29], given $0 < \epsilon < 1$, a set $\mathcal{S}$ of $2V$ points in $\mathbb{R}^{PQ}$, each with unit norm, and $K > \mathcal{O}(\frac{log(V)}{\epsilon^2})$, there exists a Lipschitz function $f : \mathbb{R}^{PQ} \rightarrow \mathbb{R}^K$ such that

$$\begin{aligned} (1-\epsilon)\|S_{n+t} - H_i^{l,m,t}\|^2 &\leq \|f(S_{n+t}) - f(H_i^{l,m,t})\|^2 \\ &\leq (1+\epsilon)\|S_{n+t} - H_i^{l,m,t}\|^2, \end{aligned} \tag{6}$$

and

$$\begin{aligned} (1-\epsilon)\|S_{n+t} + H_i^{l,m,t}\|^2 &\leq \|f(S_{n+t}) + f(H_i^{l,m,t})\|^2 \\ &\leq (1+\epsilon)\|S_{n+t} + H_i^{l,m,t}\|^2, \end{aligned} \tag{7}$$

$\forall$ S_{n+t} and $H_i^{l,m,t} \in \mathcal{S}$. Now we have:

$$\begin{aligned} & 4\langle f(S_{n+t}), f(H_i^{l,m,t}) \rangle \\ &= \|f(S_{n+t}) + f(H_i^{l,m,t})\|^2 - \|f(S_{n+t}) - f(H_i^{l,m,t})\|^2 \\ &\geq (1+\epsilon)\|S_{n+t} + H_i^{l,m,t}\|^2 - (1+\epsilon)\|S_{n+t} - H_i^{l,m,t}\|^2 \\ &= 4\langle S_{n+t}, H_i^{l,m,t} \rangle - 2\epsilon(\|S_{n+t}\|^2 + \|H_i^{l,m,t}\|^2) \\ &\geq 4\langle S_{n+t}, H_i^{l,m,t} \rangle - 4\epsilon. \end{aligned} \tag{8}$$

We can get a similar relation for opposite direction, which when combined with (8), yields the following:

$$\begin{aligned} \langle S_{n+t}, H_i^{l,m,t} \rangle - \epsilon &\leq \langle f(S_{n+t}), f(H_i^{l,m,t}) \rangle \\ &\leq \langle S_{n+t}, H_i^{l,m,t} \rangle + \epsilon. \end{aligned} \tag{9}$$

However, JL Lemma does not provide us with a embedding, f which satisfies the above relation. As discussed in [32], f can be constructed as a matrix, ϕ with size $K \times PQ$, whose entries are either independent realizations of Gaussian random variables or independent realizations of ± Bernoulli random variables. Now, if ϕ constructed as explained above is used as measurement matrix, then we can replace f in (9) by ϕ, leading us to

$$\begin{aligned}\langle S_{n+t}, H_i^{l,m,t}\rangle - \epsilon \leq \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \\ \leq \langle S_{n+t}, H_i^{l,m,t}\rangle + \epsilon.\end{aligned} \tag{10}$$

Hence, we have,

$$\begin{aligned}\sum_{t=0}^{N-1} \langle S_{n+t}, H_i^{l,m,t}\rangle - N\epsilon \leq \sum_{t=0}^{N-1} \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \\ \leq \sum_{t=0}^{N-1} \langle S_{n+t}, H_i^{l,m,t}\rangle + N\epsilon.\end{aligned} \tag{11}$$

Using equations (4) and (11), we arrive at the following desired equation.

$$c_i(l,m,n) - N\epsilon \leq \sum_{t=0}^{N-1} \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \leq c_i(l,m,n) + N\epsilon. \tag{12}$$

Now allowing for the error in correlation, we can compute the response from compressed measurements as below:

$$c_i^{comp}(l,m,n) = \sum_{t=0}^{N-1} \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle. \tag{13}$$

The above relation provides us with the 3D response volume for the test video with respect to a particular action, without reconstructing the frames of the test video. To reduce computational complexity, the 3D response volume is calculated in frequency domain via 3D FFT.

Feature vector and Classification using SVM: For a given test video, we obtain N_A correlation volumes. For each correlation volume, we adapt three level volumetric max-pooling to obtain a 73 dimensional feature vector [27]. In addition, we also compute peak-to-side-lobe-ratio for each of these 73 maxpooled values. PSR is given by $PSR_k = \frac{peak_i - \mu_i}{\sigma_i}$,where $peak_k$ is the k^{th} max-pooled value, and μ_k and σ_k are the mean and standard deviation values in its small neighbourhood. Thus, the feature vector for a given test video is of dimension, $N_A \times 146$. This framework can be used in any reconstruction-free application from compressive cameras which can be implemented using 3D correlation filtering. Here, we assume that there exists an optimal matched filter for each action and outline a way to recognize actions from compressive measurements. In the next section, we show how these optimal filters are obtained for each action.

2.2 Training filters for action recognition

The theory of training correlation filters for any recognition task is based on synthesizing a single template from training examples, by finding an optimal tradeoff between certain performance measures. Based on the performance measures, there exist a number of classes of correlation filters. A MACH filter is a single filter that encapsulates the information of all training examples belonging to a particular class and is obtained by optimizing four performance parameters, the Average Correlation Height (ACH), the Average Correlation Energy (ACE), the Average Similarity Measure (ASM), and the Output Noise Variance (ONV). Until recently, this was used only in two dimensional applications like palm print identification [31], target recognition [33] and face recognition problems [30]. For action recognition, Rodriguez et al. [22] introduced a generalized form of MACH filters to synthesize a single action template from the spatio-temporal volumes of the training examples. Furthermore, they extended the notion for vector-valued data. In our framework for compressive action recognition, we adopt this approach to train matched filters for each action. Here, we briefly give an overview of 3D MACH filters which was first described in [22].

First, temporal derivatives of each pixel in the spatio-temporal volume of each training sequence are computed and the frequency domain representation of each volume is obtained by computing a 3D-DFT of that volume, according to the following:

$$F(\mathbf{u}) = \sum_{t=0}^{N-1} \sum_{x_2=0}^{M-1} \sum_{x_1=0}^{L-1} f(\mathbf{x}) e^{(-j2\pi(\mathbf{u}\cdot\mathbf{x}))}, \tag{14}$$

where, $f(\mathbf{x})$ is the spatio-temporal volume of L rows, M columns and N frames, $F(\mathbf{u})$ is its spatio-temporal representation in the frequency domain and $\mathbf{x} = (x_1, x_2, t)$ and $\mathbf{u} = (u_1, u_2, u_3)$ denote the indices in space-time and frequency domain respectively. If N_e is the number of training examples for a particular action, then we denote their 3D DFTs by $X_i(\mathbf{u}), i = 1, 2, .., N_e$, each of dimension, $d = L \times M \times N$. The average spatio-temporal volume of the training set in the frequency domain is given by $M_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} X_i(\mathbf{u})$. The average power spectral density of the training set is given by $D_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} |X_i(\mathbf{u})|^2$, and the average similarity matrix of the training set is given by $S_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} |X_i(\mathbf{u}) - M_x(\mathbf{u})|^2$. Now, the MACH filter for that action is computed by minimizing the average correlation energy, average similarity measure, output noise variance and maximizing the average correlation height. This is done by computing the following:

$$h(\mathbf{u}) = \frac{1}{[\alpha C(\mathbf{u}) + \beta D_x(\mathbf{u}) + \gamma S_x(\mathbf{u})]} M_x(\mathbf{u}), \tag{15}$$

where, $C(\mathbf{u})$ is the noise variance at the corresponding frequency. Generally, it is set to be equal to 1 at all frequencies. The corresponding space-time domain representation $H(x, y, t)$ is obtained by taking the inverse 3D DFT of h. A filter with response volume H and parameters α, β and γ is compactly written as $\mathbf{H} = \{H, \alpha, \beta, \gamma\}$.

3 Affine Invariant Smashed Filtering

Even though MACH filters capture intra-class variations, the filters can become ineffective if viewpoints of the training examples are different or if the viewpoint of the test video is different from viewpoints of the training examples. Filters thus obtained may result in misleading correlation peaks. Consider the case of generating a filter of a translational action, walking, wherein the training set is sampled from two different views. The top row in Fig 3 depicts some frames of the filter, say 'Type-1' filter, generated out of such a training set. The bottom row depicts some frames of the filter, say 'Type-2' filter, generated by affine transforming all examples in the training set to a canonical viewpoint. Roughly speaking, the 'Type-2' filter can be interpreted as many humans walking in the same direction, whereas the 'Type-1' filter, as 2 groups of humans, walking in opposite directions. One can notice that some of the frames in the 'Type-1' do not represent the action of interest, particularly the ones in which the two groups merge into each other. This kind of merging effect will become more prominent as the number of different views in the training set increases. The problem is avoided in the 'Type-2' filter because of the single direction of movement of the whole group. Thus, it can be said that the quality of information about the action in the 'Type-2' filter is better than that in the 'Type-1' filter. As we show in experiments, this is indeed the case. Assuming that all views of all training examples are affine transforms of a canonical view, we can synthesize a MACH filter generated after transforming all training examples to a common viewpoint and avoid the merging effect. However, different test videos may be in different viewpoints, which makes it impractical to synthesize filters for every viewpoint. Hence it is desirable that a single representative filter be generated for all affine transforms of a canonical view. The following proposition asserts that, from a MACH filter defined for the canonical view, it is possible to obtain a compensated MACH filter for any affine transformed view.

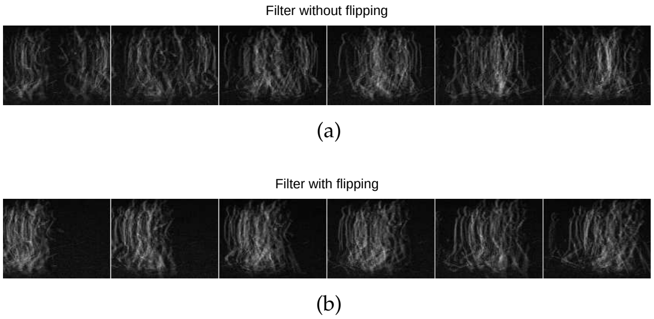


Fig. 3. a) 'Type-1' filter obtained for walking action where the training examples were from different viewpoints b) 'Type-2' filter obtained from the training examples by bringing all the training examples to the same viewpoint. In (a), two groups of human move in opposite directions and eventually merge into each other, thus making the filter ineffective. In (b), the merging effect is countered by transforming the training set to the same viewpoint.

Proposition 1: Let $\mathbf{H} = \{H, \alpha, \beta, \gamma\}$ denote the MACH filter in the canonical view, then for any arbitrary view V, related to the canonical view by an affine transformation, $[A|\mathbf{b}]$, there exists a MACH filter, $\hat{\mathbf{H}} = \{\hat{H}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$ such that: $\hat{H}(\mathbf{x_s}, t) = |\Delta|^2 H(A\mathbf{x_s} + \mathbf{b}, t)$, $\hat{\alpha} = |\Delta|^2\alpha$, $\hat{\beta} = \beta$ and $\hat{\gamma} = \gamma$ where $\mathbf{x_s} = (x_1, x_2)$ denote the horizontal and vertical axis indices and Δ is the determinant of A.

Proof: Consider the frequency domain response $\hat{h}$ for view V, given by the following.

$$\hat{h}(\mathbf{u}) = \frac{1}{(\alpha\hat{C}(\mathbf{u}) + \beta\hat{D_x}(\mathbf{u}) + \gamma\hat{S_x}(\mathbf{u}))}\hat{M_x}(\mathbf{u}). \qquad (16)$$

For the sake of convenience, we let $\mathbf{u} = (\mathbf{u_s}, u_3)$ where $\mathbf{u_s} = (u_1, u_2)$ denotes the spatial frequencies and u_3, the temporal frequency. Now using properties of the Fourier transform [34], we have,

$$\hat{M_x}(\mathbf{u_s}, u_3) = \frac{1}{N_e}\sum_{i=1}^{N_e}\hat{X_i}(\mathbf{u_s}, \mathbf{u_3})$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}.$$

Using the relation $M_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} X_i(\mathbf{u})$, we get,

$$\hat{M_x}(\mathbf{u_s}, u_3) = \frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}M_x((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}. \qquad (17)$$

Now,

$$\hat{D_x}(\mathbf{u_s}, u_3) = \frac{1}{N_e}\sum_{i=1}^{N_e}|\hat{X_i}(\mathbf{u_s}, u_3)|^2$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e}|\frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}|^2$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e}|\frac{X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}|^2 . (\because |e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}| = 1)$$

Hence, using the relation $D_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e}|X_i(\mathbf{u})|^2$, we have

$$\hat{D_x}(\mathbf{u_s}, u_3) = \frac{1}{|\Delta|^2}D_x((A^{-1})^T\mathbf{u_s}, u_3). \qquad (18)$$

Similarly, it can be shown that

$$\hat{S_x}(\mathbf{u_s}, u_3) = \frac{1}{|\Delta|^2}S_x((A^{-1})^T\mathbf{u_s}, u_3). \qquad (19)$$

Using (17), (18) and (19) in (16), we have,

$$\hat{h}(\mathbf{u}) = (e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}} M_x((A^{-1})^T\mathbf{u_s}, u_3))\Delta \frac{1}{(\hat{\alpha}|\Delta|^2\hat{C}(\mathbf{u}) + \hat{\beta}D_x((A^{-1})^T\mathbf{u_s}, u_3) + \hat{\gamma}S_x((A^{-1})^T\mathbf{u_s}, u_3)}. \quad (20)$$

Now letting, $\alpha = \hat{\alpha}|\Delta|^2$, $\beta = \hat{\beta}$, $\gamma = \hat{\gamma}$, $\hat{C}(\mathbf{u}) = C(\mathbf{u}) = C((A^{-1})^T\mathbf{u_s}, u_3))$ (since C is usually assumed to be equal to 1 at all frequencies if noise model is not available) and using (15), we have,

$$\hat{h}(\mathbf{u}) = \Delta h((A^{-1})^T\mathbf{u_s}, u_3))e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}. \quad (21)$$

Now taking the inverse 3D-FFT of $\hat{h}(\mathbf{u})$, we have,

$$\hat{H}(\mathbf{x_s}, t) = |\Delta|^2 H(A\mathbf{x_s} + \mathbf{b}, t). \quad (22)$$

Thus, a compensated MACH filter for the view V is given by $\hat{\mathbf{H}} = \{\hat{H}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$. This completes the proof of the proposition. Thus a MACH filter for view V, with parameters $|\Delta|^2\alpha$, β and γ can be obtained just by affine transforming the frames of the MACH filter for the canonical view. Normally $|\Delta| \approx 1$ for small view changes. Thus, even though in theory, $\hat{\alpha}$ is related to α by a scaling factor of $|\Delta|^2$, for small view changes, $\hat{h}$ is the optimal filter with essentially the same parameters as those for the canonical view. This result shows that for small view changes, it is possible to build robust MACH filters from a single canonical MACH filter.

Robustness of affine invariant smashed filtering: To corroborate the need of affine transforming the MACH filters to the viewpoint of the test example, we conduct the following two synthetic experiments. In the first, we took all examples in Weizmann dataset and assumed that they belong to the same view, dubbed as the canonical view. We generated five different datasets, each corresponding to a different viewing angle. The different viewing angles from $0°$ to $20°$ in increments of $5°$ were simulated by means of homography. For each of these five datasets, a recognition experiment is conducted using filters for the canonical view as well as the compensated filters for their respective viewpoints, obtained using (22). The average PSR in both cases for each viewpoint is shown in Figure 4. The mean PSR values obtained using compensated filters are more than those obtained using canonical filters.

In the second experiment, we conducted five independent recognition experiments for the dataset corresponding to fixed viewing angle of $15°$, using compensated filters generated for five different viewing angles. The results are tabulated in table 1. It is evident that action recognition rate is highest when the compensated filters used correspond to the viewing angle of the test videos. These two synthetic experiments clearly suggest that it is essential to affine transform the filters to the viewpoint of the test video before performing action recognition.

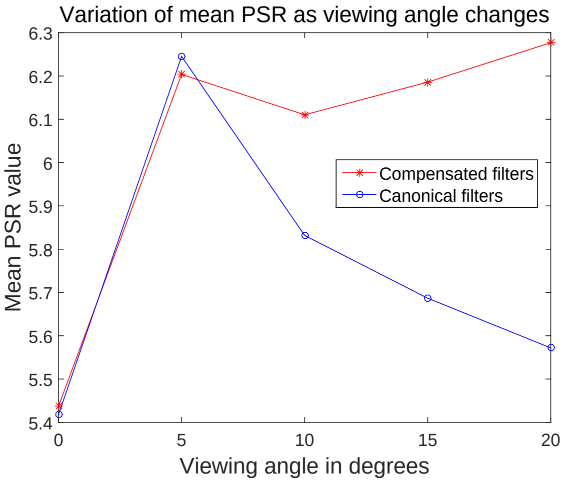


Fig. 4. The mean PSRs for different viewpoints for both canonical filters and compensated filters are shown. The mean PSR values obtained using compensated filters are more than those obtained using canonical filters, thus corroborating the need of affine transforming the MACH filters to the viewpoint of the test example.

4 Experimental Results

For all our experiments, we use a measurement matrix, ϕ whose entries are drawn from i.i.d. standard Gaussian distribution, to compress the frames of the test videos. We conducted extensive experiments on the widely used Weizmann [35], UCF sports [22], UCF50 [36] and HMDB51 [37] datasets to validate the feasibility of action recognition from compressive cameras. Before we present the action recognition results, we briefly discuss the baseline methods to which we compare our method, and describe a simple to perform action localization in those videos in which the action is recognized successfully.

Baselines: As noted earlier, this is the first paper to tackle the problem of action recognition from compressive cameras. The absence of precedent approach to this problem makes it difficult to decide on the baseline methods to compare with. The state-of-the-art methods for action recognition from traditional cameras rely on dense trajectories [10], derived using highly non-linear features, HOG [11], HOOF [12], and MBH [9]. At the moment, it is not quite clear on how to extract such features directly from compressed measurements. Due to these difficulties, we fixate on two baselines. The first baseline method is the Oracle MACH, wherein action recognition is performed as in [22] and for the second baseline, we first reconstruct the frames from the compressive measurements, and then apply the improved dense trajectories (IDT) method [10], which is the most stable state-of-the-art method, on the reconstructed video to perform action recognition. There are two approaches that one can follow to reconstruct the frames of a CS video. One of them is the naive frame-

Viewing angle	Canonical	5°	10°	15°	20°
Recognition rate	65.56	68.88	67.77	**72.22**	66.67

TABLE 1
Action recognition rates for the dataset corresponding to fixed viewing angle of 15° using compensated filters generated for various viewing angles. As expected, action recognition rate is highest when the compensated filters used correspond to the viewing angle of the test videos.

by-frame reconstruction approach, and the other one, a more sophisticated approach dubbed as video compressive sensing, involves alternating between motion estimation and motion-compensated signal recovery. We note that even the best performing video CS reconstruction algorithms [38] take about 2-3 hours to recover the video clips we deal with in this paper. We have around 7000 clips in each of the two datasets, UCF50 and HMDB51. We realized that adopting video CS reconstruction for such a large dataset is computationally infeasible. Hence, we adopt the former approach, more specifically the CoSamP algorithm [39] to reconstruct the frames of the video. We use the code made publicly available by the authors, and set all the parameters to default to obtain improved dense trajectory (IDT) features. The features thus obtained are encoded using Fisher vectors, and a linear SVM is used for classification. Henceforth, we refer this method as Recon+IDT.

Spatial Localization of action from compressive cameras without reconstruction: Action localization in each frame is determined by a bounding box centred at location (l^{max}) in that frame, where l^{max} is determined by the peak response (response corresponding to the classified action) in that frame and the size of the filter corresponding to the classified action. To determine the size of the bounding box for a particular frame, the response values inside a large rectangle of the size of the filter, and centred at l^{max} in that frame are normalized so that they sum up to unity. Treating this normalized rectangle as a 2D probability density function, we determine the bounding box to be the largest rectangle centred at l^{max}, whose sum is less than a value, $\lambda \leq 1$. For our experiments, we use λ equal to 0.7.

Computational complexity: In order to show the substantial computational savings achievable in our STSF framework of reconstruction-free action recognition from compressive cameras, we compare the computational time of the framework with that of Recon+IDT. We ran our experiments on a Intel i7 quad core machine with 16GB RAM to report the timing numbers.

Compensated Filters: In section 3, we experimentally showed that better action recognition results can be obtained if compensated filters are used instead of canonical view filters (table 1). However, to generate compensated filters, one requires the information regarding the viewpoint of the test video. Generally, the viewpoint of the test video is not known. This difficulty can be overcome by generating compensated filters corresponding to various viewpoints. In our experiments, we restrict our filters to two viewpoints described in section 3, i.e we use 'Type-1' and 'Type-2' filters.

4.1 Reconstruction-free recognition on Weizmann dataset

Even though it is widely accepted in the computer vision community that Weizmann dataset is an easy dataset, with many methods achieving near perfect action recognition rates, we believe that working with compressed measurements precludes the use of those well-established methods, and obtaining such high action recognition rates at compression ratios of 100 and above even for a simple dataset as Weizmann is not straightforward. The Weizmann dataset contains 10 different actions, each performed by 9 subjects, thus making a total of 90 videos. For evaluation, we used the leave-one-out approach, where the filters were trained using actions performed by 8 actors and tested on the remaining one. The results shown in table 2 indicate that our method clearly outperforms the Recon+IDT. It is quite evident that with full-blown frames (indicated in table 2) that Recon+IDT method performs much better than STSF method. However, at compression ratios of 100 and above, recognition rates are very stable for our STSF framework, while Recon+IDT fails completely. This is due to the fact that Recon+IDT operates on reconstructed frames, which are of poor quality at such high compression ratios, while STSF operates directly on compressed measurements. The recognition rates are stable even at high compression ratios and are comparable to the recognition accuracy for the Oracle MACH (OM) method [40].

The average time taken by STSF and Recon+IDT to process a video of size $144 \times 180 \times 50$ are shown in parentheses in table 1. Recon+IDT takes about 20-35 minutes to process one video, with the frame-wise reconstruction of the video being the dominating component in the total computational time, while STSF framework takes only a few seconds for the same

Compression factor	STSF	Recon + IDT
1	81.11 (3.22s) (OM [40], [22])	100 (3.1s)
100	81.11 (3.22s)	5.56 (1520s)
200	81.11 (3.07s)	10 (1700s)
300	76.66 (3.1s)	10 (1800s)
500	78.89 (3.08s)	7.77 (2000s)

TABLE 2
Weizmann dataset: Recognition rates for reconstruction-free recognition from compressive cameras for different compression factors are stable even at high compression factors of 500. Our method clearly outperforms Recon+IDT method and achieves a recognition rate which is comparable to the recognition rate of 81.11 in the case of Oracle MACH [40], [22].

sized video since it operates directly on compressed measurements.

Spatial localization of action from compressive cameras without reconstruction: Further, to validate the robustness of action detection using the STSF framework, we quantified action localization in terms of error in estimation of the subject's centre from its ground truth. The subject's centre in each frame is estimated as the centre of the fixed sized bounding box with location of the peak response (only the response corresponding to the classified action) in that frame as it left-top corner. Figure 5 shows action localization in a few frames for various actions of the dataset (More action localization results for Weizmann dataset can be found in supplementary material). Figure 6 shows that using these raw estimates, on average, the error from the ground truth is less than or equal to 15 pixels in approximately 70% of the frames, for compression ratios of 100, 200 and 300. It is worth noting that using our framework it is possible to obtain robust action localization results without reconstructing the images, even at extremely high compression ratios.

Experiments on the **KTH** dataset: The experimental results on the KTH dataset [41] can be found in the supplement.

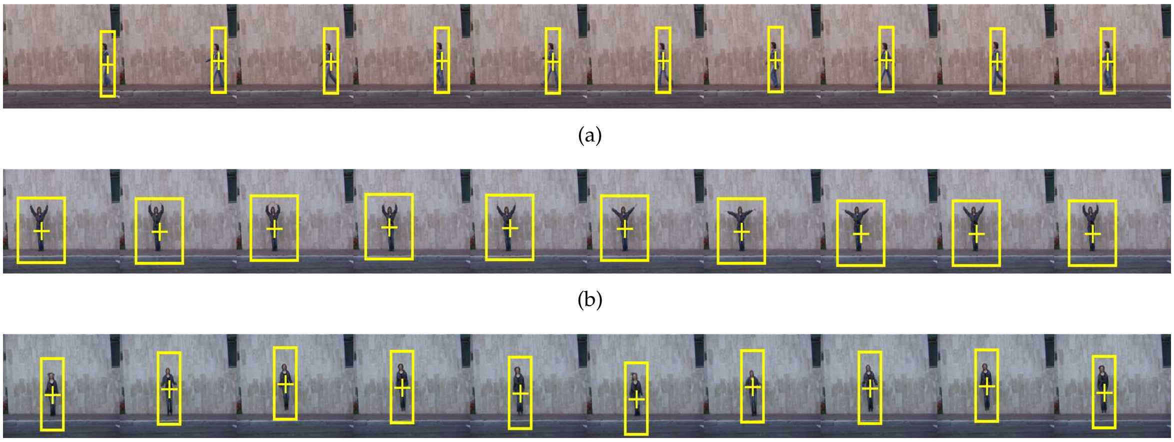
4.2 Reconstruction-free recognition on UCF sports dataset

The UCF sports action dataset [22] contains a total of 150 videos across 9 different actions. The dataset is a challenging dataset with scale and viewpoint variations. For testing, we use leave-one-out cross validation. At compression ratio of 100 and 300, the recognition rates are 70.67% and 68% respectively. The rates obtained are comparable to those obtained in Oracle MACH set-up [22] (69.2%). Considering the difficulty of the dataset, these results are very encouraging. The confusion matrix for compression ratios 100 is shown in table 3. The confusion matrix for compression ratio 300 can be found in the supplementary.

Spatial localization of action from compressive cameras without reconstruction: Figure 7 shows action localization for some correctly classified instances across various actions in the dataset, for Oracle MACH and compression ratio = 100 (More action localization results can be found in supplementary material). It can be seen that action localization is estimated reasonably well despite large scale variations and extremely high compression ratio.

4.3 Reconstruction-free recognition on UCF50 dataset

To test the scalability of our approach, we conduct action recognition on large datasets, UCF50 [36] and HMDB51 [37]. Unlike the datasets considered earlier, these two datasets have large intra-class scale variability. To account for this scale variability, we generate about 2-6 filters per action. To generate MACH filters, one requires bounding box annotations for the videos in the datasets. Unfortunately framewise bounding box annotations are not available for these two datasets. Hence, we selected 190 video clips from UCF50 dataset with 2-6 video clips per action. We manually annotated these clips with frame-wise bounding boxes. Each MACH filter is generated with just one of these videos as a training example. In total we generate 380 filters (190 canonical filters, i.e 'Type-1 filters' + 190 their flipped versions, i.e 'Type-2' filters). The UCF50 database consists of 50 actions, with around 120 clips per action, totalling upto 6681 videos. The database is divided into 25 groups with each group containing between 4-7 clips per action. We use leave-one-group cross-validation to evaluate our framework. The recognition rates at different compression ratios, and the mean time taken for one clip (in parentheses) for our framework and Recon+IDT are tabulated in table 4. Table 4 also shows the recognition rates for various state-of-the-art action recognition methods, while operating on the full-blown images, as indicated in the table by (FBI). Two conclusions follow from the table. 1) Our approach outperforms the baseline method, Recon+IDT at very high compression ratios of 100 and above, and 2) the mean time per clip is less than that for Recon+IDT method. This clearly suggests that when operating at high compression ratios, it is better to perform action recognition without reconstruction than reconstructing the frames and then applying a state-of-the-art method. The recognition rates for individual classes for Oracle MACH (OM), and compression ratios, 100 and 400 are given in table 5. The action localization results for various actions are shown in figure 8. The bounding boxes in most instances correspond to the human or the moving part of the human or the object of interest. Note how the sizes of the bounding boxes are commensurate with the area of the action in each frame. For example, for the fencing action,

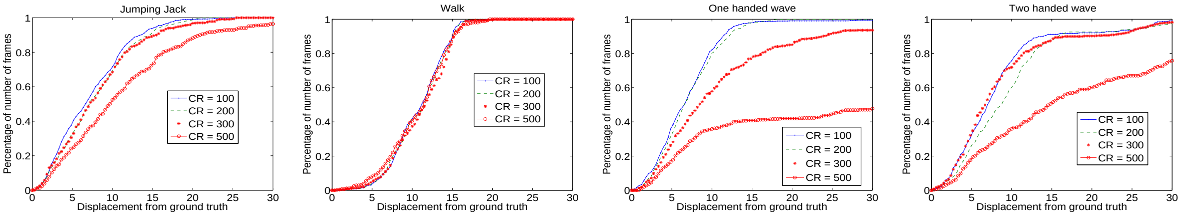


(a)

(b)

(c)

Fig. 5. Spatial localization of subject without reconstruction at compression ratio = 100 for different actions in Weizmann dataset. a) Walking b) Two handed wave c) Jump in place



(a)

(b)

(c)

(d)

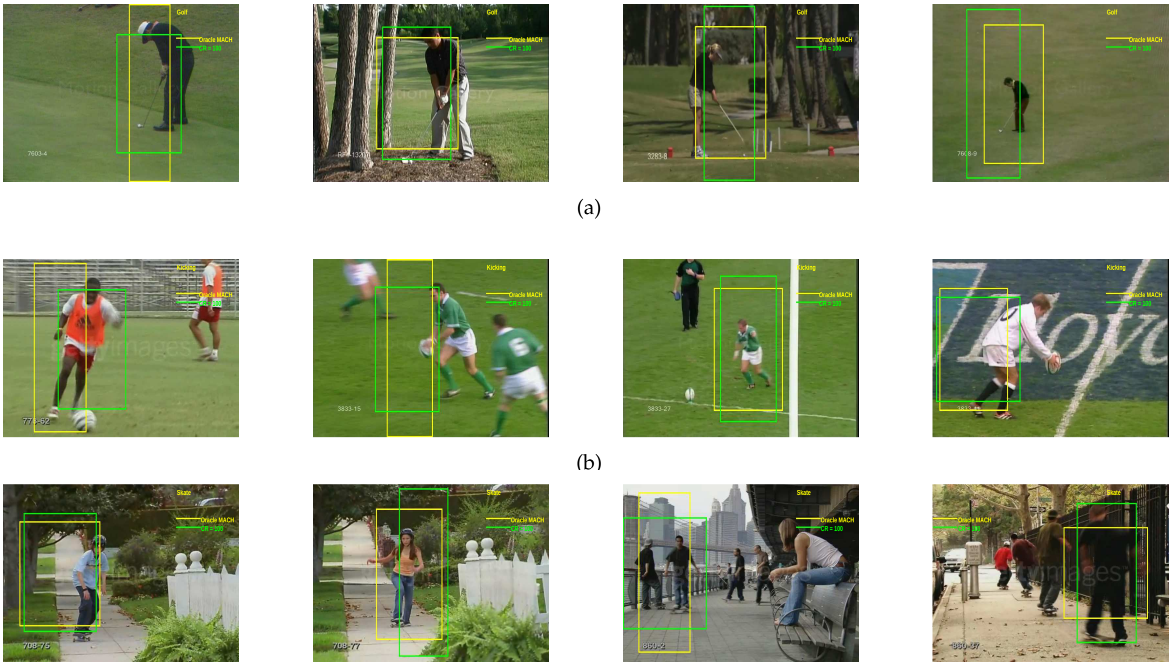
Fig. 6. Localization error for Weizmann dataset. X-axis : Displacement from ground truth. Y-axis: Fraction of total number of frames for which the displacement of subject's centre from ground truth is less than or equal to the value in x-axis. On average, for approximately 70% of the frames, the displacement of ground truth is less than or equal to 15 pixels, for compression ratios of 100, 200 and 300.

Action	Golf-Swing	Kicking	Riding Horse	Run-Side	Skate-Boarding	Swing	Walk	Diving	Lifting
Golf-Swing	**77.78**	16.67	0	0	0	0	5.56	0	0
Kicking	0	**75**	0	5	5	10	5	0	0
Riding Horse	16.67	16.67	**41.67**	8.33	8.33	0	8.33	0	0
Run-Side	0	0	0	**61.54**	7.69	15.38	7.69	7.69	0
Skate-Boarding	0	8.33	8.33	25	**50**	0	5	0	0
Swing	0	3.03	12.12	0.08	3.03	**78.79**	3.03	0	0
Walk	0	9.09	4.55	4.55	9.09	9.09	**63.63**	0	0
Diving	0	0	0	0	7.14	0	0	**92.86**	0
Lifting	0	0	0	0	0	0	0	16.67	**83.33**

TABLE 3

Confusion matrix for UCF sports database at a compression factor = 100. Recognition rate for this scenario is 70.67 %, which is comparable to Oracle MACH [22] (69.2%).

the bounding box covers both the participants, and for the playing piano action, the bounding box covers just the hand of the participant. In the case of breaststroke action, where human is barely visible, action localization results are impressive. We emphasize that action localization is achieved directly from compressive measurements without any intermediate reconstruction, even though the measurements do not bear any explicit information regarding pixel locations. We note that the procedure outlined above is by no means a full-fledged procedure for action localization and is fundamentally different from the those in [42], [43], where sophisticated models are trained jointly on action labels and the location of person in each frame, and action and its localization are determined simultaneously by solving one computationally intensive inference problem. While our method is simplistic in nature and does not always estimate localization accurately, it relies only on minimal post-processing of the correlation response, which makes it an attractive solution for action localization in resource-constrained environments where a rough estimate of action location may serve the purpose. However, we do note that action localization is not the primary goal of the paper and that the purpose of this exercise is to show that reasonable localization results directly from



(a)

(b)

(c)

Fig. 7. Reconstruction-free spatial localization of subject for Oracle MACH (shown as yellow box) and STSF (shown as green box) at compression ratio = 100 for some correctly classfied instances of various actions in the UCF sports dataset. a) Golf b) Kicking c) Skate-Boarding. Action localization is estimated reasonably well directly from CS measurements even though the measurements themselves do not bear any explicit information regarding pixel locations.

compressive measurements are possible, even using a rudimentary procedure as outlined above. This clearly suggests that with more sophisticated models, better reconstruction-free action localization results can be achieved. One possible option is to co-train models jointly on action labels and annotated bounding boxes in each frame similar to [42], [43], while extracting spatiotemporal features such as HOG3D [13] features for correlation response volumes, instead of the input video.

Method	CR = 1	CR = 100	CR =400
Our method ('Type 1' + 'Type 2')	60.86 (2300s) (OM)	54.55 (2250s)	46.48 (2300s)
Recon + IDT	91.2 (FBI)	21.72 (3600s)	12.52 (4000s)
Action Bank [27]	57.9 (FBI)	NA	NA
Jain *et al.*[44]	59.81 (FBI)	NA	NA
Kliper-Gross *et al.*[45]	72.7 (FBI)	NA	NA
Reddy *et al.*[36]	76.9 (FBI)	NA	NA
Shi *et al.*[46]	83.3 (FBI)	NA	NA

TABLE 4

UCF50 dataset: The recognition rate for our framework is stable even at very high compression ratios, while in the case of Recon + IDT, recognition rates are much lower. The mean time per clip (given in parentheses) for our method is less than that for the baseline method (Recon + IDT).

4.4 Reconstruction-free recognition on HMDB51 dataset

The HMDB51 database consists of 51 actions, with around 120 clips per action, totalling upto 6766 videos. The database is divided into three train-test splits. The average recognition rate across these splits is reported here. For HMDB51 dataset, we use the same filters which were generated for UCF50 dataset. The recognition rates at different compression ratios, and mean time taken for one clip (in parentheses) for our framework and Recon+IDT are tabulated in table 6. Table 6 also shows the recognition rates for various state-of-the-art action recognition approaches, while operating on full-blown images. The table clearly suggests that while operating at compression ratios of 100 and above, to perform action recognition, it is better to work in compressed domain rather than reconstructing the frames, and then applying a state-of-the-art method. While the recognition rates obtained using our method at different compression ratios are lower than state-of-the-art methods, they are very much comparable with Action Bank [27]. Action Bank method is the only filter based approach compared with in table 6, where linear features are extracted like in our method, whereas in the other methods highly non-linear features were extracted, which boosted action recognition accuracy substantially. The above mentioned trend can also be seen in the case of UCF50 dataset in table 4. This greatly underlines the limitations of linear features and the need to devise methods to extract non-linear features from CS videos.

4.5 Comments on computational complexity and storage

From tables 2, 4 and 6, it is evident that time taken for our framework is substantially less than that for Recon+IDT. In the case of Recon+IDT, the computational bottleneck stems from the reconstruction of

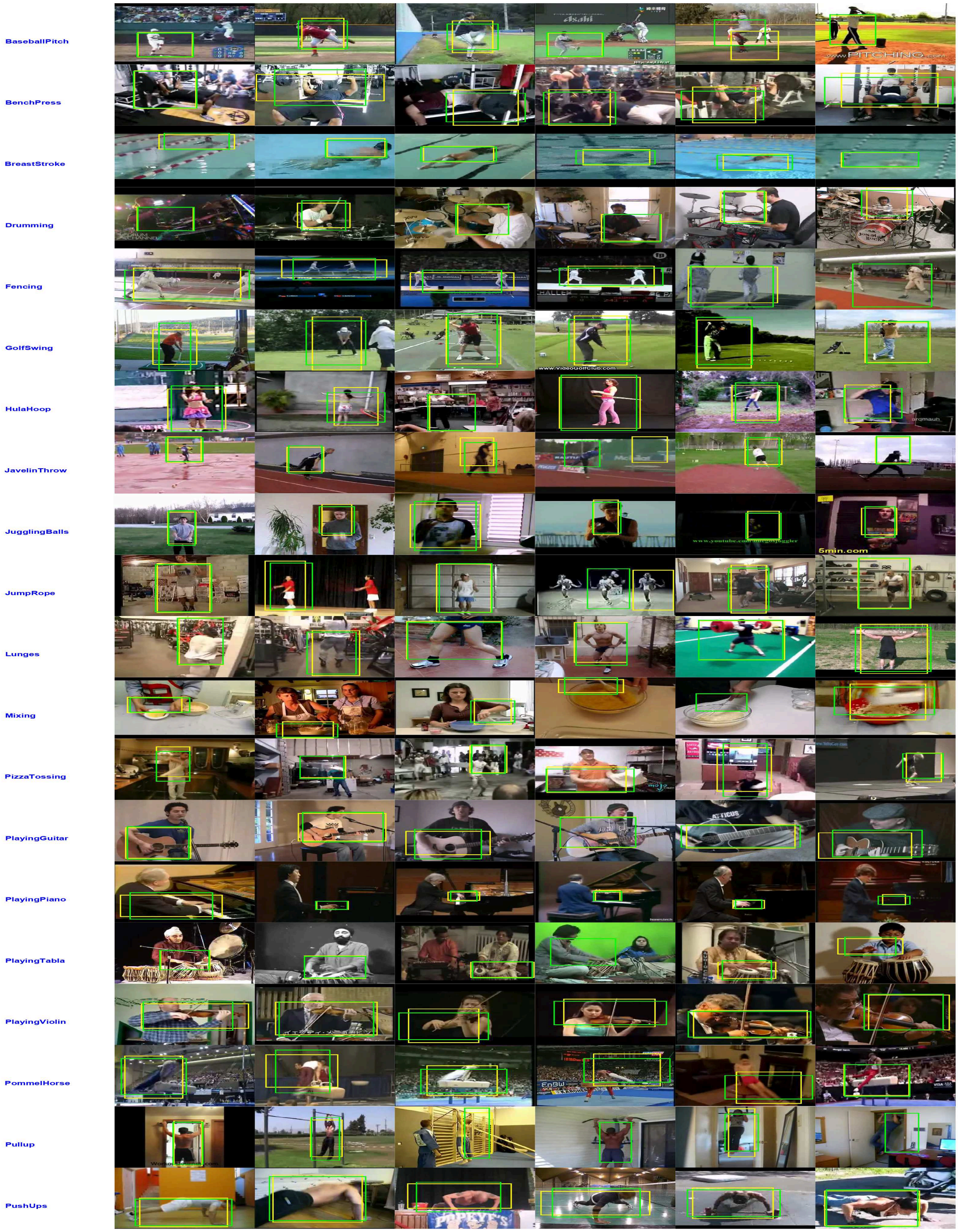


Fig. 8. Action localization: Each row corresponds to various instances of a particular action, and action localization in one frame for each of these instances is shown. The bounding boxes (yellow for Oracle MACH, and green for STSF at compression ratio = 100) in most cases correspond to the human, or the moving part. Note that these bounding boxes shown are obtained using a rudimentary procedure, without any training, as outlined earlier in the section. This suggests that joint training of features extracted from correlation volumes and annotated bounding boxes can lead to more accurate action localization results.

Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400
BaseballPitch	58.67	57.05	50.335	HorseRiding	77.16	60.4	60.4	PlayingPiano	65.71	60.95	58.1	Skiing	35.42	34.72	29.86
Basketball	41.61	38.2353	25.7353	HulaLoop	55.2	56	55.2	PlayingTabla	73.88	56.75	36.94	Skijet	44	37	29
BenchPress	80	73.75	65.63	Javelin Throw	41.0256	41.0256	32.48	PlayingViolin	59	52	43	SoccerJuggling	42.31	31.61	28.38
Biking	60	42.07	33.01	Juggling Balls	64.75	67.21	65.57	PoleVault	56.25	58.12	53.75	Swing	54.01	35.03	19.7
Billiards	94.67	89.33	79.33	JumpRope	71.53	75	74.31	PommelHorse	86.07	81.3	69.1	TaiChi	66	68	61
Breaststroke	81.19	46.53	17.82	JumpingJack	80.49	80.49	72.357	PullUp	64	59	49	TennisSwing	46.11	41.92	30.53
CleanAndJerk	56.25	59.82	41.96	Kayaking	58.6	47.14	43.12	Punch	80.63	73.12	62.5	ThrowDiscus	62.6	51.14	45
Diving	76.47	71.24	51.63	Lunges	44.68	36.17	32.62	PushUps	66.67	60.78	61.76	TrampolineJumping	45.39	28.57	18.48
Drumming	63.35	50.93	44.1	MilitaryParade	80.32	78.74	59.05	RockClimbing	65.28	58.33	63.2	VolleyBall	60.34	48.27	39.65
Fencing	71.171	64.86	62.16	Mixing	51.77	56.02	48.93	RopeClimbing	36.92	34.61	29.23	WalkingwithDog	31.71	27.64	25.4
GolfSwing	71.13	58.86	48.93	Nunchucks	40.9	34.1	31.82	Rowing	55.47	40.14	29.2	YoYo	54.69	58.59	47.65
HighJump	52.03	52.84	47.15	Pizza Tossing	30.7	33.33	22.8	Salsa	69.92	63.16	46.62				
HorseRace	73.23	66.92	59.84	PlayingGuitar	73.75	64.37	60.62	SkateBoarding	55.82	46.67	38.33				

TABLE 5

UCF50 dataset: Recognition rates for individual classes at compression ratios, 1 (Oracle MACH), 100 and 400.

Method	CR = 1	CR = 100	CR =400
Our method ('Type 1' + 'Type 2')	22.5 (2200s) (OM)	21.125 (2250s)	17.02 (2300s)
Recon + IDT	57.2 (FBI)	6.23 (3500s)	2.33 (4000s)
Action Bank [27]	26.9 (FBI)	NA	NA
Jain *et al.*[47]	52.1 (FBI)	NA	NA
Kliper-Gross *et al.*[45]	29.2 (FBI)	NA	NA
Jiang *et al.*[48]	40.7 (FBI)	NA	NA

TABLE 6

HMDB51 dataset: The recognition rate for our framework is stable even at very high compression ratios, while in the case of Recon+IDT, it is much lower.

the frames. Further, we note that for most frames, the reconstruction algorithm did not converge, owing to the high compression ratio. To avoid this, we ran the reconstruction algorithm for a fixed number of iterations. We also compared the storage and communication requirements of full blown videos and their compressed counterparts. It was observed that the raw data of a full blown video of size $240 \times 320 \times 106$ occupies 64873 KB, whereas the CS video at CR = 100 occupies 589 KB, leading to memory savings of 99.1%. Similarly, the CS video at CR = 400 occupies 147 KB, leading to memory savings of 99.77%.

5 Discussions and Conclusion

In this paper, we proposed a correlation based framework to recognize actions from compressive cameras without reconstructing the sequences. It is worth emphasizing that the goal of the paper is not to outperform a state-of-the-art action recognition system but is to build a action recognition system which can perform with an acceptable level of accuracy in heavily resource-constrained environments, both in terms of storage and computation. The fact that we are able to achieve a recognition rate of 54.55% at a compression ratio of 100 on a difficult and large dataset like UCF50 and also localize the actions reasonably well clearly buttresses the applicability and the scalability of reconstruction-free recognition in resource constrained environments. Further, we reiterate that at compression ratios of 100 and above, when reconstruction is generally of low quality, action recognition results using our approach, while working in compressed domain, were shown to be far better than reconstructing the images, and then applying a state-of-the-art method. In our future research, we wish to extend this approach to more generalizable filter-based approaches. One possible extension is to use motion sensitive filters like Gabor or Gaussian derivative filters which have proven to be successful in capturing motion. Furthermore, by theoretically proving that a single filter is sufficient to encode an action over the space of all affine transformed views of the action, we showed that more robust filters can be designed by transforming all training examples to a canonical viewpoint.

Acknowledgments

This work was supported by ONR Grant N00014-12-1-0124 sub award Z868302. The authors would like to thank Henry Braun for their useful suggestions and comments.

References

[1] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, pp. 21 – 30, 2008.

[2] M.B. Wakin, J.N. Laska, M.F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly and R.G. Baraniuk, "An architecture for compressive imaging," in *IEEE Conf. Image Process.*, 2006.

[3] V. Cevher, A. C. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Euro. Conf. Comp. Vision*, 2008.

[4] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2004.

[5] G. K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2003.

[6] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, 2001.

[7] I. Laptev, "On space-time interest points," *Intl. J. Comp. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[8] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conf.*, 2009.

[9] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2011.

[10] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *IEEE Intl. Conf. Comp. Vision*. IEEE, 2013, pp. 3551–3558.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2005, pp. 886–893.

[12] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2009.

[13] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conf.*, 2008.

[14] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, April 2011.

[15] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Sastry, "High speed action recognition and localization in compressed domain videos," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 18, no. 8, pp. 1006–1015, 2008.

[16] B. Ozer, W. Wolf, and A. N. Akansu, "Human activity detection in MPEG sequences," in *Proceedings of the Workshop on Human Motion (HUMO'00)*, ser. HUMO '00. IEEE Computer Society, 2000, pp. 61–66.

[17] A. C. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," in *Euro. Conf. Comp. Vision*, 2010.

[18] V. Thirumalai and P. Frossard, "Correlation estimation from compressed images," *J. Visual Communication and Image Representation*, vol. 24, no. 6, pp. 649–660, 2013.

[19] R. Calderbank, S. Jafarpour and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," in *Preprint*, 2009.

[20] K. Kulkarni and P. Turaga, "Recurrence textures for activity recognition using compressive cameras," in *IEEE Conf. Image Process.*, 2012.

[21] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 172 –185, 2011.

[22] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2008.

[23] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," vol. 31, no. 8, pp. 1415–1428, 2009.

[24] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2005.

[25] H. J. Seo and P. Milanfar, "Action recognition from one example," vol. 33, no. 5, pp. 867–882, 2011.

[26] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes, "Efficient action spotting based on a spacetime oriented structure representation," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2010.

[27] S. Sadanand, J. J. Corso, "Action bank: A high-level representation of activity in video," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2012.

[28] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly and R. G. Baraniuk, "The smashed filter for compressive classification and target recognition," *Computat. Imag. V*, vol. 6498, pp. 142–153, 2007.

[29] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Conference in Modern Analysis and Probability (New Haven, Conn.)*, 1982.

[30] X. Zhu, S. Liao, Z. Lei, R. Liu, and S. Z. Li, "Feature correlation filter for face recognition," in *Advances in Biometrics*. Springer, 2007, vol. 4642, pp. 77–86.

[31] P.H Hennings-Yeoman, B.V.K.V Kumar and M. Savvides, "Palmprint classification using multiple advanced correlation filters and palm-specific segmentation," *IEEE Trans. on Information Forensics and Security*, vol. 2, no. 3, pp. 613–622, 2007.

[32] A. Dimitris, "Database-friendly random projections," *Proc. ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pp. 274–281, 2001.

[33] S. Sims and A. Mahalanobis, "Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery," *Optical Engineering*, vol. 43, no. 8, pp. 1705–1711, 2004.

[34] R. Bracewell, K.-Y. Chang, A. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional Fourier transform," *Electronics Letters*, vol. 29, no. 3, p. 304, 1993.

[35] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *IEEE Intl. Conf. Comp. Vision.*, 2005.

[36] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.

[37] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *IEEE Intl. Conf. Comp. Vision.*, 2011, pp. 2556–2563.

[38] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "Cs-muvi: Video compressive sensing for spatial-multiplexing cameras," in *Computational Photography (ICCP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–10.

[39] D. Needell and J. A. Tropp, "Cosamp: iterative signal recovery from incomplete and inaccurate samples," *Communications of the ACM*, vol. 53, no. 12, pp. 93–100, 2010.

[40] S. Ali and S. Lucey, "Are correlation filters useful for human action recognition?" in *Intl. Conf. Pattern Recog*, 2010.

[41] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Intl. Conf. Pattern Recog*, 2004.

[42] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *IEEE Intl. Conf. Comp. Vision.*, 2011.

[43] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2013.

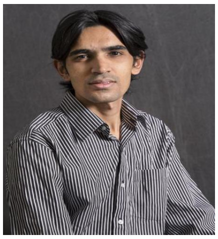
[44] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2013.

[45] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Euro. Conf. Comp. Vision*. Springer, 2012.

[46] F. Shi, E. Petriu, and R. Laganiere, "Sampling strategies for real-time action recognition," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2013.

[47] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *IEEE Conf. Comp. Vision and Pattern Recog*. IEEE, 2013.

[48] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *Euro. Conf. Comp. Vision*. Springer, 2012.



Kuldeep Kulkarni is a Phd student in the School of Electrical Engineering, and Arts, Median, Engineering at Arizona State University. He received the B.Tech degree in electrical and electronics engineering from the National Institute of Technology Karnataka, Surathkal, India, in 2009, and the M.S degree in electrical engineering from the Arizona State University in 2012. His research interests lie at the intersection of the areas of computer vision, machine learning and compressive sensing.

Pavan Turaga (S05, M09, SM14) is Assistant Professor in the School of Arts, Media, Engineering, and Electrical Engineering at Arizona State University. He received the B.Tech. degree in electronics and communication engineering from the Indian Institute of Technology Guwahati, India, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park in 2008 and 2009 respectively. He then spent two years as a research associate at the Center for Automation Research, University of Maryland, College Park. His research interests are in computer vision and computational imaging with applications in activity analysis, and dynamic scene analysis, with a focus on non-Euclidean techniques for these applications. He was awarded the Distinguished Dissertation Fellowship in 2009. He was selected to participate in the Emerging Leaders in Multimedia Workshop by IBM, New York, in 2008. He received the National Science Foundation CAREER award in 2015.