High-Speed Action Recognition and Localization in Compressed Domain Videos

Chuohao Yeo, *Student Member, IEEE*, Parvez Ahammad, *Student Member, IEEE*, Kannan Ramchandran, *Fellow, IEEE*, and S. Shankar Sastry, *Fellow, IEEE*

***Abstract*—We present a compressed domain scheme that is able to recognize and localize actions at high speeds. The recognition problem is posed as performing an action video query on a test video sequence. Our method is based on computing motion similarity using compressed domain features which can be extracted with low complexity. We introduce a novel motion correlation measure that takes into account differences in motion directions and magnitudes. Our method is appearance-invariant, requires no prior segmentation, alignment or stabilization, and is able to localize actions in both space and time. We evaluated our method on a benchmark action video database consisting of six actions performed by 25 people under three different scenarios. Our proposed method achieved a classification accuracy of 90%, comparing favorably with existing methods in action classification accuracy, and is able to localize a template video of 80 × 64 pixels with 23 frames in a test video of 368 × 184 pixels with 835 frames in just 11 s, easily outperforming other methods in localization speed. We also perform a systematic investigation of the effects of various encoding options on our proposed approach. In particular, we present results on the compression-classification tradeoff, which would provide valuable insight into jointly designing a system that performs video encoding at the camera front-end and action classification at the processing back-end.**

***Index Terms*—Action recognition, compressed domain processing, real-time video surveillance, video coding, video signal processing.**

I. Introduction

THE use of video cameras has become increasingly common as their costs decrease. In personal applications, it is common for people to record and store personal videos that comprise various actions, in part due to the widespread availability of phone cameras and cheap cameras with video recording capabilities. In security applications, multiple video cameras record video data across a designated surveillance area. A good example of this is the large network of surveillance cameras installed in London. Such proliferation of video data naturally leads to information overload. It would not only be incredibly helpful but also necessary to be able to perform rudimentary action recognition in order to assist the users in focusing their attention on actions of interest as well as allowing them to catalog their recorded videos easily.

In this paper, we formulate the problem of action recognition and localization as follows: given a query video sequence of a particular action, we would like to detect all occurrences of it in a test video, thereby recognizing an action as taking place at some specific time and location in the video. The approach should be person-independent, hence we want our method to be appearance-invariant. In a surveillance setting, it is critical to be able to respond to events as they happen. Even in a consumer application, it is desirable to minimize processing time. Therefore, we want a solution that is fast and can operate in real time.

Any practical system that records and stores digital video is likely to employ video compression such as H.263+ [2] or H.264 [3]. It has long been recognized that some of the video processing for compression can be reused in video analysis or transcoding; this has been an area of active research (see, for example, [4] and [5]) in the last decade or so. Our approach exploits this insight to attain a speed advantage.

It is reasonable to assume that a surveillance application would consist of a front-end system that records, compresses, stores, and transmits videos as well as a back-end system that processes the transmitted video to accomplish various tasks. One focus in this paper is on the action recognition task that would presumably be performed at the back-end. However, we recognize that various engineering choices, such as the choice of video coding method, made at the front-end can have an impact on the action recognition performance in the back-end. In particular, we would also like to understand how various video coding choices impact the action recognition performance of our approach.

A. Related Work

There has been a great deal of prior work in human action recognition; an excellent review of such methods has been presented by Aggarwal and Cai [6]. We are interested in approaches that work on video without relying on capturing or labeling body landmark points (see [7] and [8] for recent examples of the latter approach). Efros *et al.* [9] require the extraction of a stabilized image sequence before using a rectified optical flow-based normalized correlation measure for measuring similarity. This stabilization step required by [9] is a very challenging preprocessing step and affects the end result significantly. Shechtman and Irani [10] exhaustively test motion consistency between small space–time (ST) image intensity patches to compute a correlation measure between a query video and a test video. While their method is highly computationally intensive, they

Manuscript received October 16, 2007; revised March 5, 2008. First published February 8, 2008; current version published August 29, 2008. This work was supported in part by the National Science Foundation under Grant CCR-0330514, the Army Research Office through Award W911NF-06-1-0076, and the Agency for Science, Technology and Research, Singapore. This paper was recommended by Guest Editor I. Ahmad.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: zuohao@eecs.berkeley.edu; parvez@eecs.berkeley.edu; kannanr@eecs.berkeley.edu; sastry@eecs.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.927112

are able to detect multiple actions (similar or different) in the test video and perform localization in both space and time. Ke *et al.* [11] also use an image intensity-based approach, but apply a trained cascade of classifiers to ST volumetric features computed from image intensity. Schüldt *et al.* [12] propose an approach based on local ST features [13] in which support vector machines (SVMs) are used to classify actions in a large database of action videos that they collected. Dollar *et al.* [14] adopt a similar approach, but introduce a different spatio-temporal feature detector which they claim can find more feature points.

There has also been prior work in performing action recognition in the compressed domain. Ozer *et al.* [15] applied Principal Component Analysis (PCA) on motion vectors from *segmented* body parts for dimensionality reduction before classification. They require that the sequences must have a fixed number of frames and be *temporally aligned*. Babu *et al.* [16] trained a Hidden Markov Model (HMM) to classify each action, where the emission is a codeword based on the histogram of motion vector components of the *whole* frame. In later work [17], they extracted motion history image (MHI) and motion flow history (MFH) [18] from compressed domain features before computing global measures for classification. In [16] and [17], the use of global features precludes the possibility of localizing actions with these compressed domain methods.

B. Contributions

Our proposed method makes use of motion vector information to capture the salient features of actions which are appearance-invariant. It then computes frame-to-frame motion similarity with a novel measure that takes into account differences in both orientation and magnitude of motion vectors. The scores for each space-time candidate are then aggregated over time using a method similar to [9]. Our approach is able to localize actions in space and time by checking all possible ST candidates, much like in [10], except that it is more computationally tractable since the search space is greatly reduced from the use of compressed domain features. Our innovation lies in the ability of the proposed method to perform real-time localization of actions in space and time using a novel combination of signal processing and computer vision techniques. This approach requires no prior segmentation, no temporal or spatial alignment (unlike [9] and [15]), and minimal training. Unlike in [9], [11], [12], and [14], we also do not need to compute features explicitly; features are readily available in the compressed video data. We have to emphasize the fact that our action similarity computation is much faster than methods such as in [10], making possible applications such as content-based video organization for large-scale video databases [19].

We also study how various encoding options affect the performance of our proposed approach. This aspect is often overlooked in most other compressed domain video analysis work, in which results are typically presented only on a single choice of encoding parameters. However, we recognize that different encoding options not only affect compression performance but also influence the performance of compressed domain processing. Hence, in this study, we undertake a systematic investigation to determine the tradeoffs between compression performance and classification performance. This would be useful in understanding how best to choose encoding options to strike a good balance between compression and classification and between speed and accuracy.

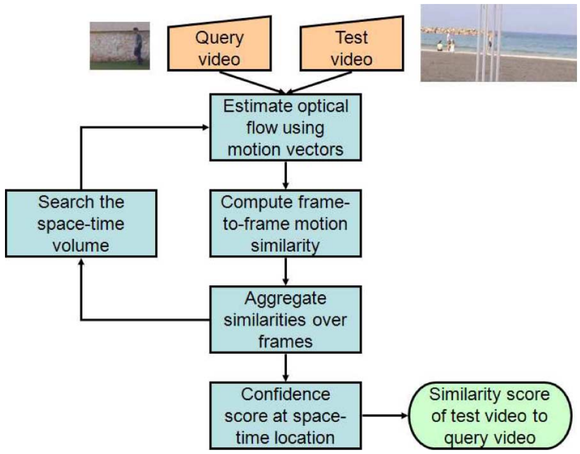

Query video
Test video
Estimate optical flow using motion vectors
Search the space-time volume
Compute frame-to-frame motion similarity
Aggregate similarities over frames
Confidence score at space-time location
Similarity score of test video to query video

Fig. 1. Flow chart of action recognition and localization method. Optical flow in the query and test videos are first estimated from motion vector information. Next, frame-to-frame motion similarity is computed between all frames of the query and test videos. The motion similarities are then aggregated over a series of frames to enforce temporal consistency. To localize, these steps are repeated over all possible ST locations. If an overall similarity score between the query and test videos is desired, a final step is performed with the confidence scores.

The remainder of this paper is organized as follows. Section II outlines our proposed method and describes each step in detail. The experimental setup and results are discussed in Section III, and we discuss the effects of different video encoding options in Section IV. We then present our concluding remarks in Section V.

II. Approach

Given a query video template and a test video sequence, we propose a compressed domain procedure to compute a score for how confident we are that the action presented in the query video template is happening at each ST location (to the nearest macroblock and frame) in the test video. The steps of the algorithm are summarized in the flow chart shown in Fig. 1. We will elaborate on each of these steps in Sections II-A–F.

Notation

In this paper, X^p denotes a video, with $p \in \{\text{test}, \text{query}\}$ referring to either the test video orthe query video. Each video X^p has T^p frames, with each frame containing $N^p \times M^p$ macroblocks. We assume that an *action* induces a motion field that can be observed as a spatio-temporal pattern; let $\vec{V}^p$ be the spatio-temporal pattern (motion field) associated with video X^p. Furthermore, $\vec{V}^p_{n,m}(i) = [V^{p,u}_{n,m}(i) \quad V^{p,v}_{n,m}(i)]$ denotes the motion vector at location (n, m) in frame i of X^p. Our working assumption is that similar actions will induce similar motion fields. We will use $(\boldsymbol{u})_+$ as a shorthand for $\max(0, \boldsymbol{u})$.

A. Estimation of Coarse Optical Flow

Motion compensation is an integral component of modern video compression technology, and motion vectors are by-products of the motion compensation process. Motion vectors are obtained from block matching and can be interpreted as crude approximations of the underlying motion field or optical flow. In addition, the discrete cosine transform (DCT) coefficients can also be used to provide a confidence measure on the estimate. We follow the approach outlined by Coimbra and Davies [20] for computing a coarse estimate and a confidence map of the optical flow. To generate the optical flow estimate, we use the following rules [20].

1) Motion vectors are normalized by the temporal distance of the predicted frame to the reference frame, and their directions are reversed if the motion vectors are forward-referencing.
2) Macroblocks with no motion vector information (e.g., macroblocks in I-frames and intra-coded macroblocks) retain the same optical flow estimate as in the previous temporal frame.
3) Macroblocks with more than one motion vector (e.g., bi-directionally predicted macroblocks in B-frames) take as the estimate a weighted average of the motion vectors, where the weights are determined by their temporal distance to the respective reference frames.

It has been recognized that optical flow estimation performance at each image location depends on the amount of texture in its local neighborhood [21]. In particular, if the local neighborhood suffers from the aperture problem, then it is likely to have an unreliable optical flow estimate. By thresholding a confidence measure derived from the DCT coefficients that measures the amount of texture in the block [20], we can filter out optical flow estimates that are likely to be unreliable. To compute the confidence measure for intra-coded macroblocks, we use [20]

$$\lambda = \frac{1}{K}\sum_{i=1}^{K}\left(F_i(0,1)^2 + F_i(1,0)^2\right)$$

where λ is the confidence measure and $F_i(u,v)$ is the 2-D DCT of the ith block $f_i(x,y)$ out of K blocks within the macroblock. Coimbra and Davies have shown that $F_i(1,0)$ and $F_i(0,1)$ can be interpreted as a weighted average of spatial gradient in the x- and y-directions, respectively [20]. For predicted macroblocks, we update the confidence map by taking a weighted average of the confidence map in the reference frame(s) as indicated by motion vector information.

By thresholding λ, we then decide whether to keep the optical flow estimate for the block or to set it to zero, hence obtaining $\vec{V}^p$. As we will show later in Section III-B, this step removes unreliable estimates and greatly improves the classification performance of our proposed algorithm.

B. Computation of Frame-to-Frame Motion Similarity

For the purpose of discussion here, both the test frame and query frame are assumed to have a spatial dimension of $N \times M$ macroblocks (the equal size restriction will be lifted later). We would like to measure the motion similarity between the motion field of the ith test frame $\vec{V}^{\text{test}}_{n,m}(i)$ and that of the jth query frame $\vec{V}^{\text{query}}_{n,m}(j)$.

One way of measuring similarity is to follow the approach taken by Efros *et al.* [9]. Each motion field is first split into nonnegative motion channels, e.g., $\left(V^{p,u}_{n,m}(i)\right)_+$, $\left(-V^{p,u}_{n,m}(i)\right)_+$, $\left(V^{p,v}_{n,m}(i)\right)_+$, and $\left(-V^{p,v}_{n,m}(i)\right)_+$ using the notation described in Section II-A. We can then vectorize these channels and stack them into a single vector $\vec{U}^p(i)$. The similarity between frame i of the test frame and frame j of the query frame, $\tilde{S}(i,j)$, is then computed as a normalized correlation

$$\tilde{S}(i,j) = \frac{\left\langle \vec{U}^{\text{test}}(i), \vec{U}^{\text{query}}(j)\right\rangle}{\left\|\vec{U}^{\text{test}}(i)\right\| \left\|\vec{U}^{\text{query}}(j)\right\|}. \tag{1}$$

We will refer to this similarity measure as *nonnegative channels normalized correlation* (NCNC).

NCNC does not take into account the differences in magnitudes of individual motion vectors. To address this, we propose a novel measure of similarity

$$\tilde{S}(i,j) = \frac{1}{Z(i,j)}\sum_{n=1}^{N}\sum_{m=1}^{M} d\left(\vec{V}^{\text{test}}_{n,m}(i), \vec{V}^{\text{query}}_{n,m}(j)\right) \tag{2}$$

where, if $\left\|\vec{V}_1\right\| > 0$ and $\left\|\vec{V}_2\right\| > 0$, then

$$\begin{aligned} d(\vec{V}_1,\vec{V}_2) &= \frac{\left(\langle \vec{V}_1, \vec{V}_2\rangle\right)_+}{\left\|\vec{V}_1\right\| \left\|\vec{V}_2\right\|} \cdot \min\left(\frac{\left\|\vec{V}_1\right\|}{\left\|\vec{V}_2\right\|}, \frac{\left\|\vec{V}_2\right\|}{\left\|\vec{V}_1\right\|}\right) \\ &= \frac{\left(\langle \vec{V}_1, \vec{V}_2\rangle\right)_+}{\max\left(\left\|\vec{V}_2\right\|^2, \left\|\vec{V}_1\right\|^2\right)} \end{aligned} \tag{3}$$

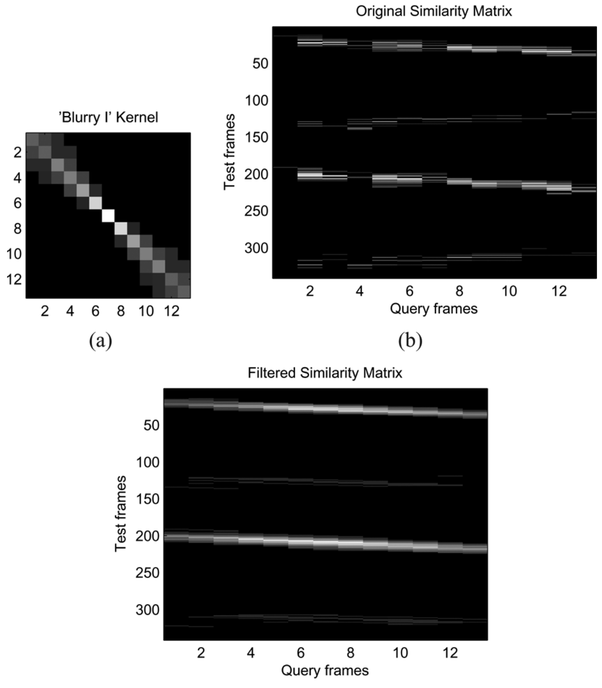
and $d(\vec{V}_1,\vec{V}_2) = 0$ otherwise. In (3), the first and second terms measure the similarity in direction and magnitude of corresponding motion vectors respectively. The normalizing factor $Z(i,j)$ in (2) is

$$Z(i,j) = \sum_{n=1}^{N}\sum_{m=1}^{M} \mathbb{1}\left[\left\|\vec{V}^{\text{test}}_{n,m}(i)\right\| > 0 \text{ or } \left\|\vec{V}^{\text{query}}_{n,m}(j)\right\| > 0\right].$$

In other words, we want to ignore macroblocks in both the query and test video which agree on having no motion. This has the effect of not penalizing corresponding zero-motion regions in both the query and test video. We term this novel measure nonzero motion block similarity (NZMS).

C. Aggregation of Frame-to-Frame Similarities

Here, we describe how to compute $\tilde{S}(i,j)$, which tells us how similar the motion fields of frame i of the test frame and frame j of the query frame are. To take temporal dependencies into account, we need to perform an aggregation step. We do this by convolving $\tilde{S}(i,j)$ with a $T \times T$ filter parameterized by α, $H_\alpha(i,j)$, to get an aggregated similarity matrix $S(i,j) = (\tilde{S} * H_\alpha)(i,j)$ [9]. $S(i,j)$ tells us how similar a T-length sequence centered at frame i of the test video is to a T-length sequence centered at frame j of the query video. $H_\alpha(i,j)$ can be interpreted as a bandpass filter that "passes" actions in the test



(a) (b)

(c)

Fig. 2. An example similarity matrix and the effects of applying aggregation. In these graphical representations, bright areas indicate a high value. (a) Aggregation kernel. (b) Similarity matrix before aggregation. (c) Similarity matrix after aggregation. Notice that the aggregated similarity matrix is less noisy than the original similarity matrix.

video that occur at approximately the same rate as in the query video. We use the following filter [9]:

$$H_\alpha(i,j) = \sum_{r\in R} e^{-\alpha(r-1)}\left(\chi(i,rj) + \chi(j,ri)\right), \quad \text{for } \frac{-T}{2} \le i,j \le \frac{T}{2}$$

where

$$\chi(u,v) = \begin{cases} 1, & \text{if } u = \text{sign}(v)\cdot\lfloor|v|\rfloor \\ 0, & \text{otherwise} \end{cases}$$

where R is the set of rates (which has to be greater than one) to allow for and $\alpha(\alpha \ge 1)$ allows us to control how tolerant we are to slight differences in rates; the higher α is, the less tolerant it is to changes in the rates of actions. Fig. 2(a) shows this kernel graphically for $\alpha = 2.0$.

Fig. 2(b) shows a pre-aggregation similarity matrix $\tilde{S}(i,j)$. Note the presence of near-diagonal bands, which is a clear indication that the queried action is taking place in those frames. Fig. 2(c) shows the post-aggregation similarity matrix, $S(i,j)$, which has much smoother diagonal bands.

We will show later in Section III-C that this aggregation step is crucial in performing action classification. However, the choice of α is not that important; experimental results show that performance is relatively stable over a range of α.

D. ST Localization

Sections II-C and D tell us how to compute an aggregated similarity between each frame of a T^{test}-frames test sequence and each frame of a T^{query}-frames query sequence, both of which are $N \times M$ macroblocks in spatial dimensions. To compute an overall score on how confident we are that frame i of the test frame is from the query sequence, we use

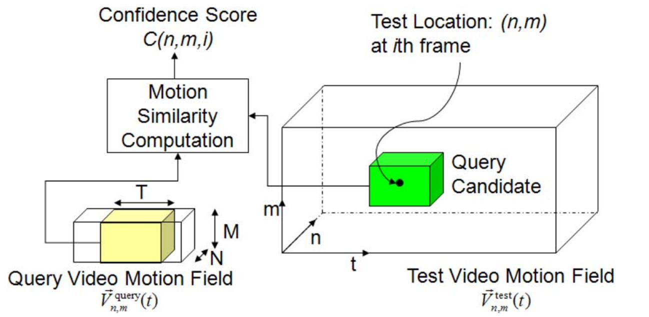


Fig. 3. Illustration of ST localization. The query video ST patch is shifted over the entire ST volume of the input video, and the similarity $C(n,m,i)$ is computed for each ST location.

$$C(i) = \max_{\substack{\max(i-(T/2),1)\le k\le \min(i+(T/2),T^{\text{test}}) \\ 1\le j\le T^{\text{query}}}} S(k,j). \tag{4}$$

Maximizing $S(k,j)$ over j of the query video allows us to pick up the best response that a particular frame of the test video has to the corresponding frame in the query video. We also maximize $S(k,j)$ over k in a T-length temporal window centered at i. The rationale is that if a T-length sequence centered at frame k of the test video matches well with the query video, then all frames in that T-length sequence should also have at least the same score.

The above steps can be easily extended to the case where the test video and query video do not have the same spatial dimensions. In that case, as proposed by Shechtman and Irani [10], we simply slide the query video template over all possible spatial-temporal locations (illustrated in Fig. 3) and compute a score for each ST location using (4). This results in an action confidence volume $C(n,m,i)$ that represents the score for the (n,m) location of the ith frame of the test video. A high value of $C(n,m,i)$ can then be interpreted as the query action being likely to be occurring at spatial location (n,m) in the ith frame.

While this exhaustive search seems to be computationally intensive, operating in the compressed domain allows for a real-time implementation.

E. Video Action Similarity Score

Given $C(n,m,i)$, we can compute a nonsymmetric similarity, $\rho(X^{\text{test}}, X^{\text{query}})$, of the test video to the query video by using

$$\rho(X^{\text{test}}, X^{\text{query}}) = \frac{1}{L}\sum_{i=1}^{T_{\text{test}}} \eta(i)\left(\max_{n,m} C(n,m,i)\right)$$

where the normalization factor L is given by

$$L = \sum_{i=1}^{T_{\text{test}}} \eta(i)$$

and $\eta(i)$ is an indicator function which returns one if at least T frames in the $(2T+1)$-length temporal neighborhood centered

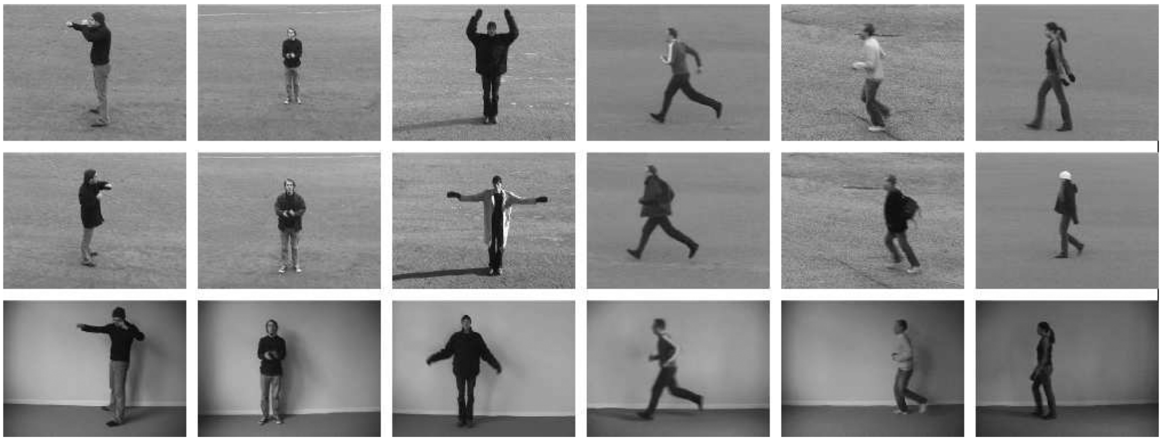


Fig. 4. Snapshots of frames from action videos in database [12]. From left to right: boxing, handclapping, handwaving, running, jogging, and walking. From top to bottom: outdoors environment, outdoors with different clothing environment, and indoors environment. The subjects performing each action are the same across the different environments.

at frame i have significant motion and returns zero if otherwise, i.e.,

$$\eta(i) = \mathbb{1}\left[\sum_{j=i-T}^{i+T} \mathbb{1}\left[Q(j) \geq \delta\right] \geq T\right]$$

and the fraction of significant motion vectors in frame j, $Q(j)$, is given by

$$Q(j) = \frac{\sum_{n=0}^{N^{\text{test}}-1} \sum_{m=0}^{M^{\text{test}}-1} \mathbb{1}\left[\left\|\vec{V}_{n,m}^{\text{test}}(j)\right\| > \epsilon\right]}{N^{\text{test}} \cdot M^{\text{test}}}.$$

A frame is asserted to have significant motion if at least δ proportion of the macroblocks have reliable motion vectors (reliable in the sense defined in Section II-B) of magnitude greater than ϵ, i.e., $Q(j) \geq \delta$.

III. Experimental Results

We evaluate our proposed algorithm on a comprehensive database compiled by Schüldt *et al.* [12].[1] As illustrated in Fig. 4, their database captures six different actions (boxing, handclapping, handwaving, running, jogging and walking), performed by 25 people, over four different environments (outdoors, outdoors with scale variations, outdoors with different clothes, and indoors). Since our system was not designed to handle scale-varying actions, we considered only the three environments that do not have significant scale variations.

To evaluate performance, within each environment, we perform a leave-one-out full-fold cross validation, i.e., to classify each video in the dataset, we use the remaining videos that are not of the same human subject as the training set. This will improve the statistical significance of our results given the limited number of videos in the dataset. To perform classification, we simply use nearest neighbor classification (NNC) by evaluating the video action similarity score (see Section II-F) with each of the videos in the training set.

TABLE I
Confusion Matrix Using NZMS

	Box	Hc	Hw	Run	Jog	Walk
Boxing	0.86	0.07	0.05	0.00	0.00	0.01
Handclapping	0.03	0.89	0.08	0.00	0.00	0.00
Hand**w**aving	0.00	0.04	0.96	0.00	0.00	0.00
Running	0.00	0.00	0.00	0.79	0.21	0.00
Jogging	0.00	0.00	0.00	0.01	0.97	0.01
Walking	0.00	0.00	0.00	0.00	0.07	0.93

TABLE II
Confusion Matrix Using Normalized Correlation [9]

	Box	Hc	Hw	Run	Jog	Walk
Boxing	0.86	0.00	0.01	0.00	0.00	0.12
Handclapping	0.43	0.32	0.24	0.00	0.00	0.00
Hand**w**aving	0.01	0.01	0.97	0.00	0.00	0.00
Running	0.00	0.00	0.00	0.97	0.03	0.00
Jogging	0.00	0.00	0.00	0.21	0.79	0.00
Walking	0.00	0.00	0.00	0.00	0.61	0.39

In our experiments, we used $\delta = (1/30)$, $\epsilon = 0.5$ pels/frame, $\alpha = 2.0$, and $T = 17$. For comparison, we also tested both NCNC [(1)] and NZMS [(2)] when computing frame-to-frame motion similarity.

A. Classification Performance

The action classification confusion matrix for our algorithm when using NZMS is shown in Table I while that using NCNC [9] is shown in Table II. Each entry of the matrix gives the fraction of videos of the action corresponding to its row that was classified as an action corresponding to the column. Using the proposed NZMS, our overall percentage of correct classification is 90%. As a comparison against state-of-the-art methods that work in the pixel domain, we note here for reference that Schüldt *et al.* [12], Dollar *et al.* [14], and Ke *et al.* [11] report classification accuracies of 72%, 81%, and 63%, respectively, on the same dataset. While the methodology and classification methods used in these works differ, our results compare very favorably, even though we use compressed domain features and a very simple classifier.

[1] [Online]. Available: http://www.nada.kth.se/cvap/actions/

TABLE III
CLASSIFICATION PERFORMANCE WITH AND WITHOUT THRESHOLDING CONFIDENCE MAP

Method	With thresholding	Without thresholding
NZMS	90.0%	81.2%
NCNC	71.7%	72.5%

TABLE IV
CLASSIFICATION PERFORMANCE WITH VARYING α

α	Classification performance
1.0	88.2%
2.0	90.0%
3.0	91.0%
4.0	90.8%
No aggregation	62.5%

Looking at the confusion matrices, we see that our proposed NZMS measure vastly outperforms NCNC. This is due to the fact that our measure looks at each corresponding pair of macroblocks separately instead of looking across all of them. NZMS also considers both differences in motion vector orientations and norms, and ignores matching zero-motion macroblocks.

Using NZMS, most of the confusion is between "Running" and "Jogging," with a significant proportion of "Jogging" videos being erroneously classified as "Running." Looking at the actual videos visually, we found it hard to distinguish between some "Running" and "Jogging" actions. In fact, there are certain cases where the speed of one subject in a "Jogging" video is faster than the speed of another subject in a "Running" video.

B. Performance Gain From Thresholding Optical Flow Confidence Map

Table III shows the effects of thresholding on action classification performance using our proposed approach. By removing noisy estimates of the optical flow, we are able to achieve a 10% gain in classification performance when using NZMS as the motion similarity measure.

C. Effect of α Variation on Classification Performance

To understand the effect of α on classification, we ran an experiment using NZMS with varying values of α. Table IV shows the results of this experiment. We see that the classification performance is relatively stable over a range of α. More importantly, it is also clear that the aggregation step described in Section II-D is critical for action classification.

D. Localization Performance

Unlike most other methods, with the notable exception of [10] and [11], we are able to localize an action in space and time as well as detect multiple and simultaneously occurring activities in the test video. Fig. 5 shows an example (the "beach" test sequence and walking query sequence from Shechtman and Irani [10]) that demonstrates our algorithm's ability to detect multiple people walking in the test video. We emphasize that we only use a single template video of a person walking to localize walking actions in the test video. Since our algorithm is not appearance-based, there is no problem with using a query video of one person on a test video containing other people.

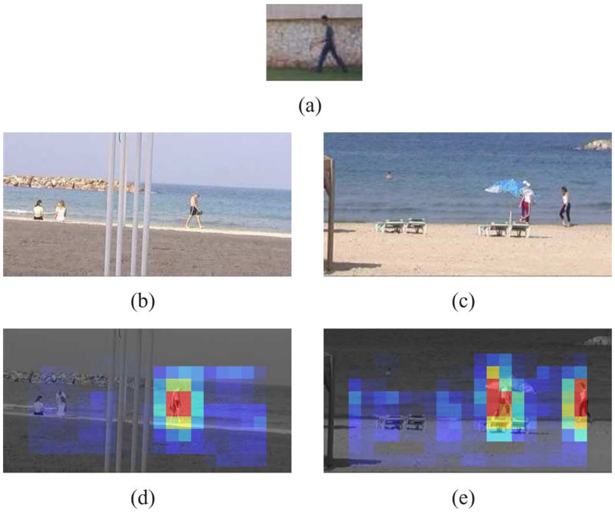


Fig. 5. Localization results. (a) A frame from the query video. (b) An input video frame with one person walking. (c) An input video frame with two people walking. (d) Detection of one person walking. (e) Detection of two people walking. The false color in (d) and (e) denotes detection responses, with blue and red indicating a low or high response, respectively.

In the test sequence, there are both static background clutter, such as people sitting and standing on the beach, and dynamic background clutter, such as sea waves and a fluttering umbrella. This background is very different from that in the query sequence. Since the spatio-temporal motion field of background motion such as sea waves is different from that of walking, it is not picked up by our algorithm. No special handling of the background motion is necessary.

E. Computational Costs

On a Pentium-4 2.6-GHz machine with 1 GB of RAM, it took just under 11 s to process a test video of 368 × 184 pixels with 835 frames on a query video that is 80 × 64 pixels with 23 frames. We extrapolated the timing reported in [10] to this case; it would have taken about 11 h. If their multigrid search was adopted, it would still have taken about 22 min. Our method is able to perform the localization, albeit with a coarser spatial resolution, up to three orders of magnitude faster. On the database compiled in [12], each video has a spatial resolution of 160 × 120 pixels and has an average of about 480 frames. For each environment, we would need to perform 22500 cross comparisons. However, each run took an average of about 8 h. In contrast, [10] would have taken an extrapolated run time of three years.

IV. EFFECTS OF VIDEO ENCODING OPTIONS

In the experiments described in the previous section, we have used input video compressed with MPEG [22], with a group-of-pictures (GOP) size of 15 frames and a GOP structure of I-B-B-P-B-B-, where "I" refers to an intra-frame, "P" refers to a predicted-frame, and "B" refers to a bi-directionally predicted-frame. It would be interesting to see if there is any discernible difference when different encoding options, such as GOP size, GOP structure, or the use of half-pel or quarter-pel motion estimation, are used. In addition, while MPEG uses 16 × 16 pixel macroblock as the basis of motion compensation,

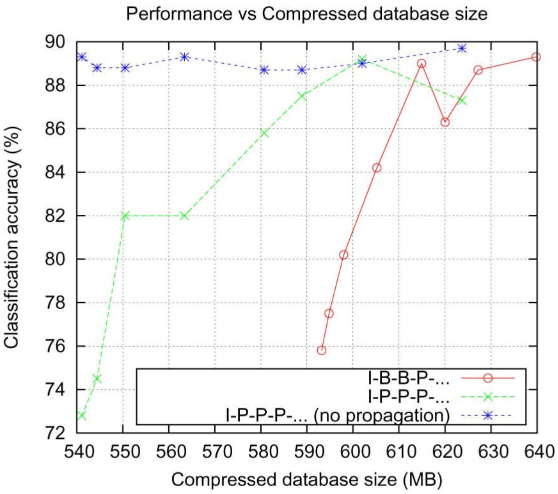


Fig. 6. Effect of varying GOP size on classification performance and compression performance. In general, increasing GOP size results in decreasing classification performance. Also, having no B-frames in the GOP structure offers a better compression-classification tradeoff. The fairly constant performance of the scheme using I-P-P-P-… with no texture propagation error indicates that the main source of performance degradation with increasing GOP size is due to propagation errors in computing block texture.

newer encoding standards such as H.263+ and H.264 allow the use of smaller block sizes [2], [3].

These experiments would be useful for a systems engineer in choosing a video encoder and its encoding options. While storage space and video quality are important considerations, it would be helpful to know if sacrificing a little compression performance would yield large performance gains in surveillance tasks such as action detection.

In the experiments below, we have used the publicly available "FFMPEG" video encoder.[2] When applicable, we will describe the encoder options and specify the actual flags used with FFMPEG in parentheses. Unless otherwise mentioned, the encoding options used are that the MPEG-4 video codec is used ("-vcodec mpeg4"), the output video is of similar quality to the input video ("-sameq"), and the "AVI" container format is used.

A. *GOP Size and Structure*

We first look at how varying GOP size and structure affects classification performance. We consider two commonly used GOP structure, I-B-B-P-B-B- ("-bf 2") and I-P-P-P-P-P-. We also look at a variety of GOP sizes {9,12,15,18,30,60,120,240} ("-g [GOP size]"). By looking at how classification performance varies with compression performance, we can get an idea of what tradeoffs are possible by varying GOP parameters when performing video encoding. In these experiments, the output video quality is kept relatively similar over all GOP size and structure.

It should be expected, and is in fact the case, that the larger the GOP size, the smaller the compressed videos, since predicted frames such as P- and B-frames can be more efficiently compressed than I-frames. The results in Fig. 6 further shows that, in general, increasing GOP size also results in decreasing classification performance. This could be due to the fact that the update of the confidence measure computed as in Section II-B suffers from error propagation with each P-frame. To test out this hypothesis, we also ran experiments where the confidence measure is computed from the DCT of the actual decoded frame pixels instead. Looking at the curve for the I-P-P-P-… GOP structure with no texture propagation error, we see that the classification accuracy is indeed fairly constant over a wide range of GOP size. This confirms that the main source of performance degradation with increasing GOP size is due to the propagation errors in computing the confidence measure.

Fig. 6 also shows that, for the most part, the I-P-P-P-… GOP structure offers a better classification–compression tradeoff than the I-B-B-P… GOP structure. There are two possible reasons for this. First, because of the complexity of articulated motion, B-frames are unable to provide any substantial compression gains over P-frames, while suffering from overhead. Hence, the I-B-B-P-… structure, for the same GOP size, actually performs worse in terms of compression performance. Second, the I-B-B-P-… structure introduces inaccuracy into the optical flow estimation process. The P-frames are spaced three frames apart, and hence its estimated motion is actually over three temporal frames and not over one frame.

The experiments in this section seem to suggest that, if action classification is an important factor in determining encoding options, then no B-frames should be used in the encoding. This also has other advantages such as simpler encoders and decoders requiring less frame buffer memory. Further, if we used the confidence measure as computed in Section II-B, the GOP size should not be too large. A GOP size of 12, 15, or 18 seems to offer a good balance between compression and action classification. There might also be other factors in determining GOP size, however, such as ease of random access and error resilience.

B. *Quarter-Pel Accuracy Motion Estimation*

In MPEG, motion estimation was carried out to half-pel accuracy. It was found that better motion compensation is possible with a further increase in accuracy to quarter-pel [3], [23]. This motivates us to investigate if an increase in motion estimation accuracy ("-qpel 1") would also translate into better action classification performance.

Fig. 7 shows that using quarter-pel accuracy in motion estimation does not actually improve the classification–compression tradeoff. There are two main reasons for this. First, we observe that on this set of action videos, for the same GOP size, using quarter-pel accuracy actually performs worse than half-pel accuracy in terms of compression performance. This could be due to the storage overhead of motion vectors with increased accuracy. Second, quarter-pel accuracy does not translate into better action classification performance.

C. *Block Size in Motion Compensation*

As mentioned earlier, newer encoding standards have the option of allowing smaller block sizes to be used in motion compensation [2], [3]. We compare the effect of forcing smaller blocks in motion compensation ("-mv4 1") on both action classification performance and compression performance. In this set of experiments, we used a GOP structure of I-B-B-P-.

[2] [Online]. Available: http://ffmpeg.mplayerhq.hu/

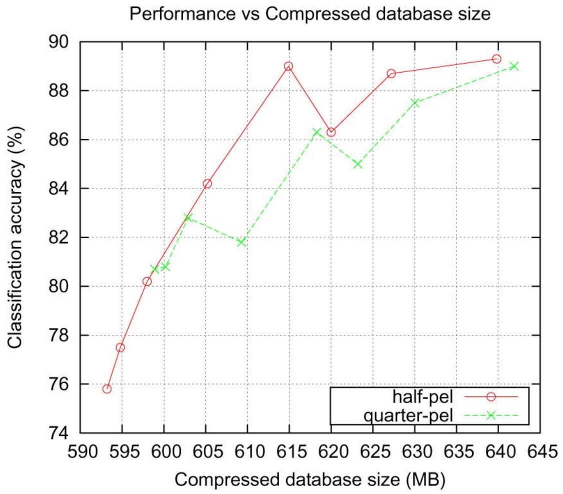


Fig. 7. Effect of quarter-pel accuracy motion estimation on classification performance and compression performance. There seems to be no significant improvement in the compression-classification tradeoff by using motion estimation with quarter-pel accuracy instead of half-pel accuracy.

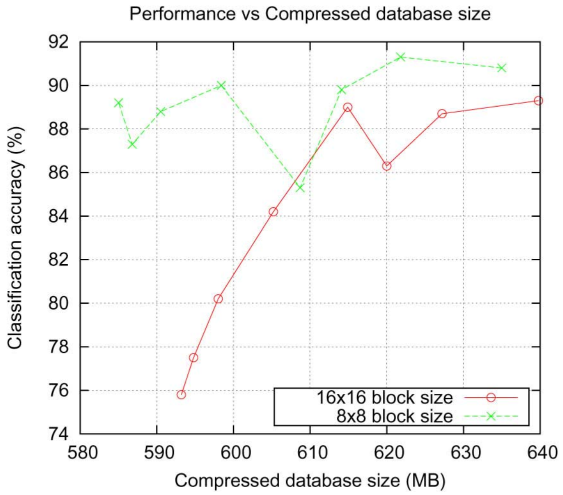


Fig. 8. Effect of using different block sizes in motion compensation on classification performance and compression performance. Using a smaller block size results in a better compression–classification tradeoff, but this has to be weighed against the resulting increase in computational time.

Fig. 8 shows that using smaller blocks in motion compensation does result in a better performance-versus-compression tradeoff. Smaller blocks allows for a more refined motion compensation and prediction, hence resulting in better compression performance. At the same time, with higher resolution motion vectors, action classification performance also improves. Of course, while using smaller blocks for motion compensation improves the tradeoff, it has to be weighted by the increase in computation time. In our experiments, increasing the motion estimation resolution by 2 in each dimension resulted in about 5 times increase in run-time.

V. Conclusion

We have designed, implemented, and tested a system for performing action recognition and localization by making use of compressed domain features such as motion vectors and DCT coefficients that can be obtained with minimal decoding. The low computational complexity of feature extraction and the inherent reduction in search space make real-time operation feasible. We combined existing tools in a novel way in the compressed domain for this purpose and proposed NZMS, which is a novel frame-to-frame motion similarity measure. Our classification results compare favorably with existing techniques [11], [12], [14] on a publicly available database, and the computational efficiency of our approach is significantly less than existing action localization methods [10].

Our experimental results provide justification for the engineering choices made in our approach. In particular, we showed the value of filtering motion vectors with low texture and of aggregating frame-to-frame similarities. We also systematically investigated the effects of various encoding options on the action classification performance of our proposed approach. The results showed that, for action videos, using a GOP structure with only P-frames results in a better compression–classification tradeoff. We also found that, while a larger GOP size might result in a lower classification performance, it is mostly due to the effects of drift in computing block texturedness. Thus, a simple extension for improving classification performance in videos with large GOP size, if memory constraints permit, is to perform full decoding of every frame and to use the decoded pixels at shorter regular intervals to update the confidence map. We found that quarter-pel accuracy in motion estimation does not appear to provide any benefits. While using smaller blocks in motion compensation does lead to better action classification and compression performance, the increased computational time of both encoding and action classification should be taken into account.

In this study, we have used a very simple classifier, i.e., NNC, which still has given a very good performance. For further improvement in classification, we can use more sophisticated classifiers such as SVMs; on the same dataset, Dollar *et al.* have shown that using SVMs result in a slight improvement over NNC [14].

For future work, we plan to extend our system to adopt a hierarchical approach which would allow us to approach the spatial resolution of existing pixel-domain methods at lower computational cost. By leveraging the ability of state-of-the-art encoders such as H.264 to use smaller blocks in motion compensation, motion vectors at resolutions of up to 4×4 pixels block can be obtained. The algorithm can first perform action recognition at the coarsest level, i.e., 16×16 pixels macroblock, and then perform a progressively finer level search in promising regions. Furthermore, using the motion vectors of 4×4 pixels block as an initial estimate also allows the computation of dense optical flow at lower cost, hence enabling the progressive search to proceed to pixel level granularity.

One current limitation of our approach is that, while it is robust to small variations in spatial scale, it is not designed to handle large-spatial-scale variations or differences in spatial scales between the query and test videos. We would like to explore a truly scale-invariant approach in future work. A possibility is to apply our method at different resolutions in parallel; this can be done naturally with the hierarchical extension described earlier. Parallelizing this scale-space search could lead to significant gains in performance while being scale-invariant.

While we present results on a benchmark dataset widely used for evaluating activity recognition algorithms [11], [12], [14], it would be interesting to consider data with other actions and containing more varied backgrounds as part of future work. For example, the BEHAVE project, which has the objective of automatically detecting anomalous or criminal behavior from surveillance videos, has publicly available datasets.[3] One interesting approach uses optical flow information to identify such behavior [24]; it would be useful to see how our method, which uses only motion vectors, compares with the former, which uses optical flow. While we consider single person actions, detecting multiparty activities such as greeting or fighting is also a potential area of further investigation [24], [25].

Another interesting angle to consider is the type of motion estimation used at the encoder. Rate-distortion (RD) optimization is commonly performed in sophisticated video encoders to seek an optimum tradeoff between compression and reconstruction quality [26]. It has also been used in the motion compensation process to reduce the rate used for coding motion vectors [27], [28]. This has the effect of smoothing the motion vector field which can be interpreted as a de-noising process. We hypothesize that this has a positive influence on the compression–classification tradeoff, but this would have to be verified.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that have helped improve the quality of this manuscript.

References

[1] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Sastry, "Compressed domain real-time action recognition," in *Proc. IEEE Workshop on Multimedial Signal Processing*, Victoria, BC, Canada, Oct. 2006, pp. 33–36.

[2] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "$H.263+$: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 849–866, Nov. 1998.

[3] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[4] S.-F. Chang, "Compressed-domain techniques for image/video indexing and manipulation," in *Proc. IEEE Int. Conf. Image Process.*, 1995, pp. 314–317.

[5] S. Wee, B. Shen, and J. Apostolopoulos, "Compressed-domain video processing," Hewlett-Packard, Tech. Rep. HPL-2002-282, 2002.

[6] J. Aggarwal and Q. Cai, "Human motion analysis: A review," in *IEEE Proc. Nonrigid Articulated Motion Workshop*, 1997, pp. 90–102.

[7] A. Yilma and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 150–157.

[8] V. Parameswaran and R. Chellappa, "Human action-recognition using mutual invariants," *Comput. Vis. Image Understanding*, vol. 98, no. 2, pp. 294–324, 2005.

[9] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 726–733.

[10] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, San Diego, CA, Jun. 2005, pp. 405–412.

[11] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 166–173.

[12] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. Int. Conf. Pattern Recogn.*, Cambridge, U.K., Aug. 2004, pp. 32–36.

[13] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 432–439.

[14] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop Visual Surveillance Perform. Eval. Tracking Surveillance*, 2005, pp. 65–72.

[15] B. Ozer, W. Wolf, and A. N. Akansu, "Human activity detection in MPEG sequences," in *Proc. IEEE Workshop Human Motion*, Austin, TX, Dec. 2000, pp. 61–66.

[16] R. V. Babu, B. Anantharaman, K. Ramakrishnan, and S. Srinivasan, "Compressed domain action classification using HMM," *Pattern Recogn. Lett.*, vol. 23, no. 10, pp. 1203–1213, Aug. 2002.

[17] R. V. Babu and K. R. Ramakrishnan, "Compressed domain human motion recognition using motion history information," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, pp. 321–324.

[18] J. Davis and A. Bobick, "The representation and recognition of action using temporal templates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 1997, pp. 928–934.

[19] P. Ahammad, C. Yeo, K. Ramchandran, and S. Sastry, "Unsupervised discovery of action hierarchies in large collections of activity videos," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2007, pp. 410–413.

[20] M. T. Coimbra and M. Davies, "Approximating optical flow within the MPEG-2 compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 103–107, Jan. 2005.

[21] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[22] D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[23] T. Wedi and H. Musmann, "Motion-and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, Jul. 2003.

[24] E. Andrade, S. Blunsden, and R. Fisher, "Hidden Markov models for optical flow analysis in crowds," in *Proc. 18th Int. Conf. Pattern Recogn.*, 2006, vol. 1, pp. 460–463.

[25] M. Ryoo and J. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, 2006, vol. 2, pp. 1709–1718.

[26] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Jun. 1998.

[27] G. Sullivan and R. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. IEEE Global Telecommun. Conf.*, 1991, vol. 3, pp. 85–90.

[28] M. Chen and A. Willson, Jr, "Rate-distortion optimal motion estimation algorithms for motion-compensated transform video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 147–147, Apr. 1998.

Chuohao Yeo (S'05) received the S.B. degree in electrical science and engineering and the M.Eng. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002. He is currently working toward the Ph.D. degree in electrical engineering and computer science at the University of California (UC), Berkeley.

From 2001 to 2002, he was a Research Assistant with the Research Laboratory of Electronics, MIT. He was a Research Engineer with the Institute for Infocomm Research, Singapore, in the summer of 2004 and was an Engineering Intern with Omnivision Technologies, Sunnyvale, CA, in the summer of 2005. Since 2005, he has been a Graduate Student Researcher with the Berkeley Audio Visual Signal Processing and Communication Systems Laboratory, UC, Berkeley. His research interests include image and video processing and communications, distributed source coding, computer vision, and machine learning.

Mr. Yeo is a student member of SPIE. He was a recipient of the Singapore Government Public Service Commission Overseas Merit Scholarship from 1998–2002 and a recipient of Singapore's Agency for Science, Technology and Research Overseas Graduate Scholarship since 2004. He received a Best Student Paper Award at SPIE VCIP 2007.

[3][Online]. Available: http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/

Parvez Ahammad (S'99) received the B.E. degree in electronics and communication from Osmania University, Hyderabad, India, the M.S. degree in electrical engineering from the University of Central Florida, Orlando, and the M.S. degree in computer science and the Ph.D. degree in electrical engineering and computer sciences from the University of California (UC), Berkeley.

He was a Research Engineer Intern with the Nokia Research Center in the summer of 2001 and with Logitech Inc., Fremont, CA, in the summer of 2006. He has been a Graduate Student Researcher with the UC Berkeley Robotics and Intelligent Machines Laboratory since 2003. His research interests are in computer vision and signal processing, particularly in their application to biological problems (high-throughput imaging and analysis), and camera networks (image and video understanding).

Mr. Ahammad is a student member of ACM and SPIE.

Kannan Ramchandran (S'92–M'93–SM'98–F'05) received the Ph.D. degree in electrical engineering from Columbia University, New York, in 1993.

He is a Professor with the Electrical Engineering and Computer Science Department, University of California (UC), Berkeley, where he has been since 1999. From 1993 to 1999, he was on the Faculty of the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign (UIUC). Prior to that, he was a Member of the Technical Staff with AT&T Bell Laboratories from 1984 to 1990. His current research interests include distributed systems theory for large-scale networks, video communication over wireless networks, multi-user information theory, security, and multiscale statistical signal processing and modeling.

Dr. Ramchandran was the recipient of the Elaihu I. Jury Award in 1993 at Columbia University (for the best doctoral thesis in the area of systems, signal processing, or communications), the National Science Foundation CAREER Award in 1997, the Office of Naval Research and Army Research Office Young Investigator Awards in 1996 and 1997 respectively, the Henry Magnusky Scholar Award from the Electrical and Computer Engineering Department at UIUC (chosen to recognize excellence among junior faculty), and the Okawa Foundation Prize from the Electrical Engineering and Computer Science Department at UC Berkeley in 2001. He was the corecipient of two Best Paper Awards from the IEEE Signal Processing Society (1993 and 1997). He serves on numerous technical program committees for premier conferences in image, video, and signal processing, communications, and information theory. He has been a member of the technical committees of the IEEE Image and Multidimensional Signal Processing Committee and the IEEE Multimedia Signal Processing Committee and has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

S. Shankar Sastry (F'94) received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1977, and the M.S. degree in electrical engineering and computer science, the M.A. degree in mathematics, and the Ph.D. degree in electrical engineering and computer science from University of California (UC), Berkeley, in 1979, 1980, and 1981, respectively.

He was an Assistant Professor with the Massachusetts Institute of Technology, Cambridge, from 1981 to 1983 and joined the faculty of UC Berkeley in 1983. An internationally recognized expert on embedded and autonomous software, he has an exceptional background in technology research, spearheading projects to improve the nation's cyber security and network infrastructure. He has held leadership positions in the federal government and on the Berkeley campus, most recently as director of the Center for Information Technology Research in the Interest of Society (CITRIS).

Dr. Sastry is a member of the National Academy of Engineering and the American Academy of Arts and Sciences. He was the recipient of the National Science Foundation Presidential Young Investigator Award and the Eckman Award of the American Automatic Control Council. He was also the recipient of the President of India Gold Medal, the IBM Faculty Development Award, an honorary degree from Harvard, and the distinguished Alumnus Award of the Indian Institute of Technology in 1999.