Event-Based Analysis of Video *

Lihi Zelnik-Manor **Michal Irani**
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel
Email: {lihi,irani}@wisdom.weizmann.ac.il

Abstract

Dynamic events can be regarded as long-term temporal objects, which are characterized by spatio-temporal features at multiple temporal scales. Based on this, we design a simple statistical distance measure between video sequences (possibly of different lengths) based on their behavioral content. This measure is non-parametric and can thus handle a wide range of dynamic events. Having an event-based distance measure between sequences, we use it for a variety of tasks, including: (i) event-based search and indexing into long video sequences (for "intelligent fast forward"), (ii) temporal segmentation of long video sequences based on behavioral content, and (iii) clustering events within long video sequence into event-consistent sub-sequences (i.e., into event-consistent "clusters"). These tasks are performed without prior knowledge of the types of events, their models, or their temporal extents.

Our simple event representation and associated distance measure supports event-based search and indexing even when only one short example-clip is available. However, when multiple example-clips of the same event are available (either as a result of the clustering process, or supplied manually), these can be used to refine the event representation, the associated distance measure, and accordingly the quality of the detection and clustering process.

1 Introduction

Dynamic events can form a powerful cue for analysis of video information, including event-based video indexing, browsing, clustering, and segmentation. Analysis of events [27, 2, 10, 16, 15, 7, 18, 11, 21] has primarily focused on the recognition of sets of predefined events or actions, or assumed restricted imaging environments. For example, Ju et al. model and recognize articulated motions [10], Black and Yacoob treat facial expressions [2], and the approaches of Polana and Nelson [18], Cutler and Davis [5], Liu and Picard [11] and of Saisan et al. [21] are designed to detect periodic activities. These methods propose elegant approaches for capturing the important characteristics of these events/actions by specialized parametric models with a small number of parameters. These parametric models usually give rise to high-quality recognition of the studied actions. The construction of these parametric models is usually done via an extensive learning phase, where many examples of each studied action are provided (often manually segmented and/or manually aligned).

However, real-world applications are unlikely to be restricted to recognition of pre-studied carefully

*This work was supported by the European Commission Project IST-2000-26001 VIBES and by the Israeli Ministry of Science Grant no. 1229.

modelled events. When dealing with general video data (e.g., movies), often there is no prior knowledge about the types of events in the video sequence, their temporal and spatial extent, or their nature (periodic/non-periodic). A desired application might be for the user who is viewing a movie (e.g., a sports movie), to point out an interesting video segment which contains an event of interest (e.g., a short clip which shows a tennis serve), and request the "system" to fast-forward to the next clip (or find all clips) where a "similar" event occurs. We refer to this as *"event-based video indexing"*, or *"Intelligent Fast-Forward"*. Such applications require developing a notion of event-based similarity which is based on a less-specialized (but also less restrictive) approach to event modelling.

Approaches for modelling events in non-parametric ways in the form of dynamic textures have been previously suggested (e.g., [4, 18, 25, 21, 1]). Some for synthesis, a few for recognition (but usually recognition of periodic events). Our approach bears resemblance to this category of methods, but (i) applies to both periodic and non-periodic events, and (ii) is insensitive to photometric and spatial changes (e.g., due to different worn clothes, etc.).

We regard an event as a stochastic temporal process, where local space-time measurements at multiple temporal scales are taken as samples of the stochastic process, and are used to construct an empirical distribution associated with this event. Our particular choice of space-time measurements preserves the temporal variations while being insensitive to spatial changes, and applies to both periodic and non-periodic events. The distance between empirical distributions provides a simple statistical distance measure between video sequences (possibly of different lengths) based on their behavioral content. This measure is non-parametric and can thus handle a wide range of dynamic events. It may not be optimal for a specific action, but allows for general event-based analysis of video information containing unknown event types.

Having an event-based distance measure between sequences, we can use it for temporal segmentation of long video sequences based on behavioral content. Alternatively, we can use it for clustering events within long continuous video sequences without prior knowledge of the types of events, their models, or their temporal extent. An outcome of such a clustering process is again a temporal segmentation of a long video sequence into event-consistent sub-sequences but also yields their grouping into event-consistent clusters. The proposed temporal segmentation is based on similarity of behavioral content, and is thus fundamentally different from the standard temporal segmentation of video into "scenes" or "shots" (e.g., [14, 29, 20, 13]). An approach to behavioral-based temporal segmentation of video was proposed by Rui & Ananadan [19]. However, unlike [19], our approach provides temporal segmentation into richer non-atomic actions and events. For example, in a clip showing a tennis player striking the ball, the approach of [19] will segment the strike into its fragments (e.g., backward movement of the arm, forward movement of the arm, forward step, etc.), whereas our approach can treat the entire tennis strike as a single event.

While our event-based distance measure is inferior in accuracy to the more specialized (but more restricted) parametric models (e.g., [27, 2, 10]), it can be refined with the gradual increase in knowledge about the underlying data. This gives rise to a stratified approach to event-based detection and indexing: When only *one* short example clip of the event-of-interest is available, our simple and crude measure can be used for event-based indexing and detection. However, when multiple example clips of the same event are available (either as a result of the clustering process, or pointed-out manually), these can be used to refine our event representation, the associated distance measure, and the quality of the detection and clustering process.

A preliminary version of this paper appeared in [28].

2 What is an Event?

Events are long-term temporal objects, which usually extend over tens or hundreds of frames. Polana and Nelson [18] separated the class of temporal events into three groups and suggested separate approaches for modelling and recognizing each: (i) *temporal textures* which have indefinite spatial and temporal extent (e.g., flowing water), see [17] , (ii) *activities* which are temporally periodic but spatially restricted (e.g., a person walking), see [18], and (iii) *motion events* which are isolated events that do not repeat either in space or in time (e.g., smiling). In this paper we refer to *temporal events* as all of the above, and would like to treat all of them within a single framework.

Temporal objects (events) and *spatial objects* bare many similarities as well as differences. Spatial objects are usually characterized by multiple spatial scales [8, 3, 22]. Similarly, temporal objects (events) are characterized by multiple temporal scales. For example, in a sequence of a walking person, the high temporal resolutions will capture the motion of the arms and legs, whereas the low temporal resolutions will mostly capture the gross movement of the entire body.

However, there is a major difference between spatial and temporal objects. Due to the perspective nature of the projection in the spatial dimension, a spatial object may appear at different spatial scales in different images (e.g., depending on whether it is imaged from far or near). In contrast, a temporal event is always characterized by the same temporal scales in all sequences. This is due to the "orthographic" nature of the projection along the temporal dimension (which is simply the temporal sampling at constant frame rate). For example, a single step of a walking person, viewed by two different video cameras of the same frame rate, will extend over the same number of frames in both sequences, regardless of the internal or external camera parameters. Hence, the same event will be captured at the same temporal scales in different sequences, even when viewed from different distances, different viewing positions, or at different zooms. This observation has motivated us to represent and analyze events by performing measurements and comparing them at corresponding temporal scales across different sequences.

3 Event Representation

Based on the above observations, local space-time measurements at multiple *temporal scales* of the video sequence are taken as samples of a stochastic temporal process (the event), and are used to construct an empirical distribution associated with this event at each temporal scale. Two events are considered similar if they could have been generated by the same stochastic process, i.e., if their empirical distributions at corresponding temporal scales are similar. This is explained next.

For obtaining measurements at multiple temporal scales we first construct a *temporal pyramid* of the entire video sequence by blurring and sub-sampling the sequence along the temporal direction only (see Figure 1). The temporal pyramid of a sequence S is thus a pyramid of sequences $S^1(=S), S^2, \ldots, S^L$, where the image frames in all the sequences are of the same size, and each sequence S^l has half the number of frames of the higher resolution sequence S^{l-1}. For example, given a sequence $S = S^1$ of 600 frames of size 200×300, we blur it using a 1D gaussian filter (we used a 5 tap filter) along the temporal dimension. We then sub-sample along the temporal dimension by a factor of 2 resulting in a sequence S^2 of 300 frames of the same spatial dimensions as the frames in S^1 (200×300). Similarly S^3 will have 150 frames of size 200×300 and so forth. We usually use 3 or 4 temporal scales (i.e., L=3 or 4).

For each level (sequence) S^l in the temporal pyramid, we estimate the local space-time intensity gradient (S^l_x, S^l_y, S^l_t) at each space-time point (x, y, t) (See Figure 2). The gradient is normal to the local spatio-temporal surface generated by the event in the space-time sequence volume (at temporal resolution l). The gradient *direction* captures the local surface orientation, which depends mostly on

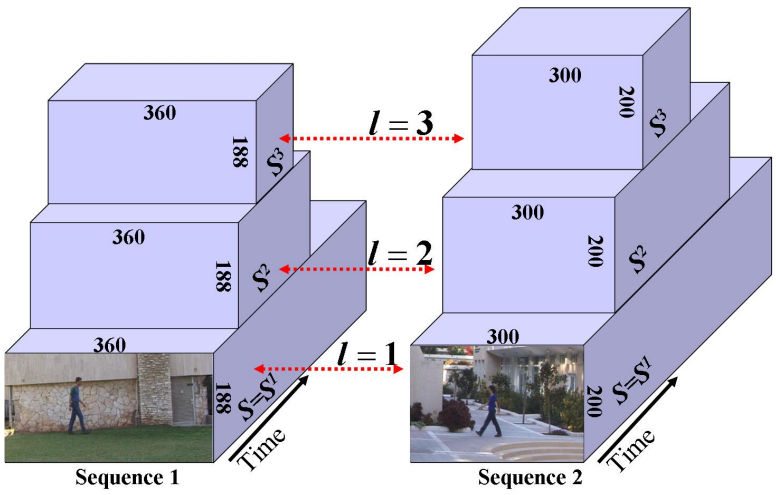

Figure 1. The Temporal Pyramid: *The temporal pyramid is constructed by blurring (convolving with a 5-tap 1D gaussian) and sub-sampling (by a factor of 2) the sequence along the temporal direction only. The image frames in all pyramid levels have the same spatial dimensions but each level has half the number of frames of the previous pyramid level (resulting in different* temporal *resolutions). Two sequences (e.g. "sequence 1" and "sequence 2") are compared by comparing measurements across corresponding temporal scales ($l = 1, 2, 3$). This can be done even if the two sequences have different frame sizes and different temporal lengths.*

the local behavioral properties of the moving object, while its *magnitude* depends primarily on the local photometric properties of the moving object and is affected by its spatial appearance (e.g., color, texture of clothes, illumination, etc.). To preserve the orientation (behavioral) information alone and eliminate as much of the photometric component as possible (the magnitude), we normalize the spatio-temporal gradients to be of length 1. To be invariant to negated contrasts between foreground and background (e.g., a person wearing dark/light clothes against a light/dark background) and to the direction of action (e.g., walking right-to-left or left-to-right), we further take the absolute value of the normalized space-time gradients. Our local space-time measurements are therefore:

$$(N_x^l, N_y^l, N_t^l) = \frac{(|S_x^l|, |S_y^l|, |S_t^l|)}{\sqrt{(S_x^l)^2 + (S_y^l)^2 + (S_t^l)^2}} \tag{1}$$

We associate with each event a set of $3L$ empirical distributions $\{p_k^l\}$, one for each component of the space-time measurements $(k = x, y, t)$ at each temporal scale $(l = 1, \ldots, L)$. These empirical distributions capture the statistics of the spatio-temporal shape generated by the event. The empirical distribution p_k^l of measurements N_k^l is represented by a discrete smoothed histogram h_k^l whose integral is normalized to 1. For example, when using 3 temporal scales an event is represented by a set of 9 one-dimensional histograms: $\{h_x^1, h_y^1, h_t^1, h_x^2, h_y^2, h_t^2, h_x^3, h_y^3, h_t^3\}$. The normalization of the histograms gives rise to similar histograms (i.e., statistics) for sequences displaying the same event, even when the two sequences are of different temporal lengths or of different spatial sizes.

We ignored all space-time points (x, y, t) for which the temporal derivative is below some threshold, thus performing the statistics (computing the histograms) mostly on spatio-temporal points which participate in the event. This step can be regarded as a very rough spatial segmentation, and was sufficient for our purposes (this was used for all the results shown in this work).

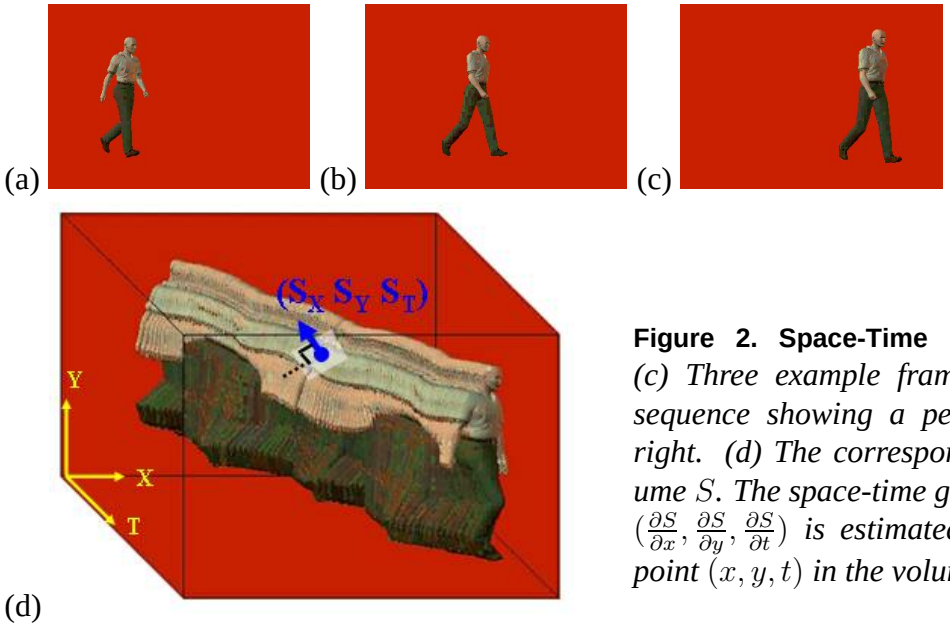

Figure 2. Space-Time Measurements: *(a)-(c) Three example frames from an example sequence showing a person walking left-to-right. (d) The corresponding space-time volume S. The space-time gradient $(S_x, S_y, S_t) = (\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial t})$ is estimated at each space-time point (x, y, t) in the volume S.*

To illustrate this, Figure 3 shows the empirical distributions (histograms) h_k^l of one component of space-time measurements N_k^l ($k = t, l = 1$) for 6 different video clips. Three clips (Figure 3.a,e,f) show "walking" activities performed by different people wearing different clothes, viewed from different viewing angles, from different distances and with different backgrounds. The other three clips (Figure 3.b,c,d) show the same person (wearing the same clothes, same background, etc.) performing different activities ("jog", "wave" and "roll"). The distributions of all the 'walking' clips (marked in blue solid lines) are much closer to each other than to those corresponding to the other activities (marked in different colors and different line types).

Unlike Chomat & Crowley [4], which measure motion features at multiple *spatial* scales, our measurements are performed at multiple *temporal* scales. We therefore capture *temporal textures* as opposed to "moving spatial textures" which are captured by [4]. Although [18, 25] used the term "temporal textures", in fact they did not measure texture properties at multiple temporal scales. [23] used the term "video textures" with a different meaning - for synthesizing video by temporally shuffling frames. Saisan et al. [21] model "dynamic textures" in video sequences as auto-regressive processes and perform recognition by examining the auto-regressive model parameters. This yields good results for sequences with temporal stationarity (e.g., waves, steam, etc.) but cannot be applied to non-stationary dynamic events such as a single tennis stroke. Spatio-temporal video textures have been previously used by [1] for video synthesis. While in video synthesis it is important to preserve both the spatial and the temporal properties of the texture (in order to generate a long realistic looking sequence from a short clip), in event recognition and detection we do not want to be sensitive to the spatial texture, only to the temporal texture properties. Insensitivity to spatial texture is necessary in order to detect different people wearing different clothes as performing the same dynamic operations. The spatio-temporal measurements we use in this work (unlike those of [1]) are relatively insensitive to the changes in spatial properties of the acting person or of the background.

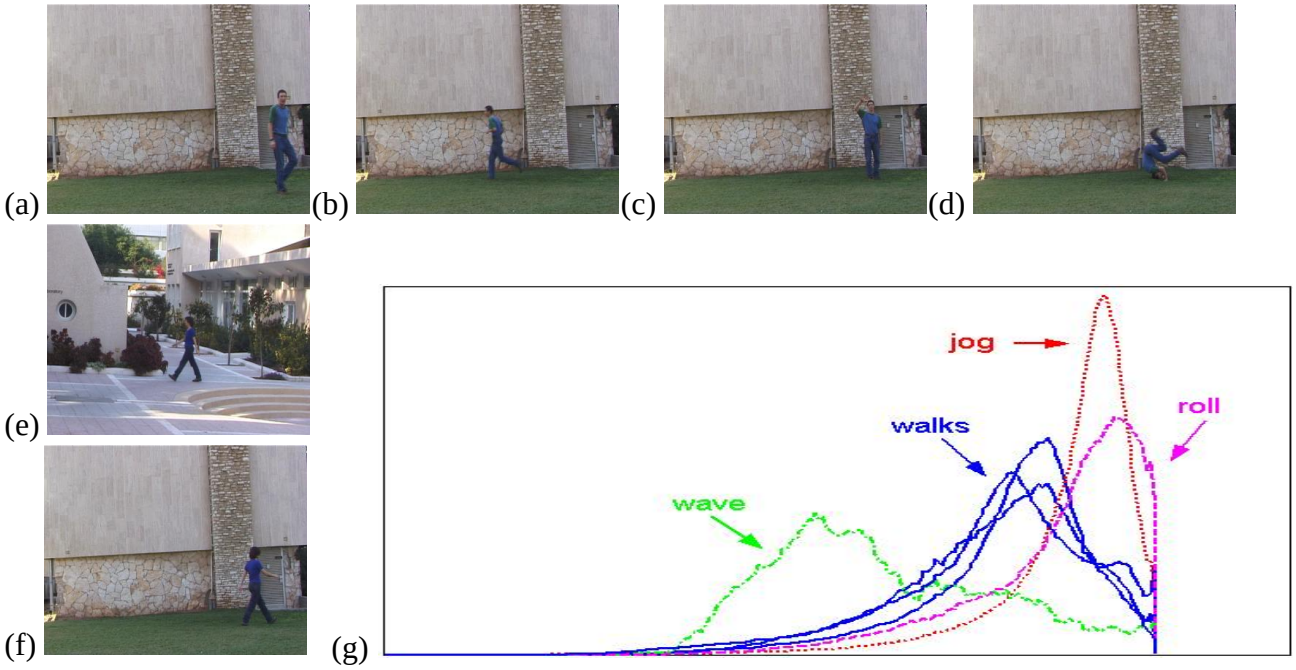

Figure 3. Distributions of Space-Time Measurements: *(a),(e),(f) Example frames from three clips showing walking activities performed by different people wearing different clothes, viewed from different viewing angles, from different distances and with different backgrounds. (b),(c),(d) Example frames from three clips showing the same person performing different activities (while wearing the same clothes, same background, etc.). (g) The empirical distribution h_t^1 of N_t^1 for the six different clips. The distributions of all the "walking" clips (marked in blue solid lines) are much closer to each other than to those corresponding to the other activities ("jog", "roll", and "wave", marked in different colors and different line types).*

4 An Event-Based Distance Measure

To measure the "behavioral" distance between two sequences (S_1 and S_2) we measure the distances between corresponding empirical distributions of all the components of the space-time measurements at all temporal scales using χ^2 divergence, and add these to obtain a single (squared) distance measure between the two sequences:

$$D^2 = \frac{1}{3L} \sum_{k,l,i} \frac{\left[h_{1k}^l(i) - h_{2k}^l(i)\right]^2}{h_{1k}^l(i) + h_{2k}^l(i)} \tag{2}$$

Prior smoothing of the histograms decreases the sensitivity of the χ^2 test to small local miss-matches between the histograms $\{h_{1k}^l\}$ and $\{h_{2k}^l\}$. Note, however, that the sequences S_1 and S_2 need *not* be pre-aligned.

In general, enforcing simultaneous occurrence of space-time measurements at multiple temporal scales should provide a better distance measure than Eq. (2). However, this requires the use of multi-dimensional histograms (e.g., [22]). These are computationally intensive and memory-consuming (e.g., for $k = 3$, $L = 4$, and assuming 256 bins for each histogram dimension, the size of the multi-dimensional histogram is 256^{12}). Instead we use single-dimensional histograms and require the simultaneous occurrence of *distributions* of space-time measurements at multiple temporal scales. The use of single-dimensional histograms reduces the data size to $12 \cdot 256$, which is easy to manage and computation-

ally fast. However, by using the single-dimensional distributions we implicitly assume independence between the components of the space-time measurements of each spatio-temporal point, i.e., we assume that $(N^1_{x_i}, N^1_{y_i}, N^1_{t_i}, \ldots, N^L_{x_i}, N^L_{y_i}, N^L_{t_i})$ of the spatio-temporal point (x_i, y_i, t_i) are independent (an assumption which is usually inaccurate). However, as can be seen from our results, requiring the simultaneous occurrence of *distributions* of space-time measurements $\{h^l_k\}$ was found to give satisfactory results.

Figure 4 shows the effectiveness of the distance measure of Eq. (2) for event detection and indexing based on a single example clip. The video sequence was several-minutes long (approx. 6000 frames) recorded outdoors by a stationary video camera (see Figure 4.a - 4.f). The sequence contains four types of frequently occurring activities: walking, jogging, hand-waving, and walking-in-place (performed by different people of both genders wearing different clothes for different lengths of time), and single occurrences of several other activities (e.g., rolling, and other free activities). Most of the walking is performed parallel to the image plane, but several parts include walking in slightly diagonal directions and some on snake-like paths. A short clip (64 frames long) where a single person walked left-to-right was selected and compared against a sliding window continuously shifted across the entire long video sequence. Small values in the graph of Figure 4.g indicate temporal regions in the long sequence with high similarity (small distance) to the example clip, while large values indicate low similarity (large distance). The blue bars on the time axis mark the video segments where a person actually walked (i.e., manually marked ground-truth).

Figure 5 further illustrates the effectiveness of the distance measure of Eq. (2) by applying the event detection scheme to a video sequence showing a person performing three types of actions: 'punch', 'kick' and 'duck' in an arbitrary order. A short clip where the person performed a single punch was selected and compared against a sliding window continuously shifted across the entire long video sequence. Here again, small values in the graph of Figure 5.d indicate temporal regions in the long sequence with high similarity (small distance) to the example clip, while large values indicate low similarity (large distance). The blue bars on the time axis mark the video segments in which the person actually punched (i.e., manually marked ground-truth).

Our simplistic (but general) event-based representation and distance measure are probably inferior in accuracy to the more sophisticated (but more restricted) parametric approaches (e.g., [27, 2, 10]). However, they can handle a wide range of unknown events and actions. Furthermore, as was shown in Figure 5, a *single* example clip of an event suffices to generate our event representation and measure its behavioral distance from other video clips. In Section 7 we discuss how an improved representation and distance measure can be obtained with the gradual increase of information about the underlying data, i.e., when more example clips of the same event are available.

5 Event-Based Temporal Segmentation

Having an event-based distance measure between video sequences, we can use it for temporal segmentation of long video sequences according to their behavioral content. This is different from the standard temporal segmentation into "scenes" or "shots" (e.g., [14, 29, 20, 13]), which is based on scene-cut or shot-cut detection. We next suggest a scheme for temporal segmentation of long video sequences, which is fast and simple to implement.

We use a sliding temporal window to compare every sub-sequence of length T to its consecutive sub-sequence of length T (i.e., we compare the sub-sequence extending from frame $f-T+1$ to frame f with its consecutive sub-sequence extending from frame $f+1$ to frame $f+T$). For each pair of consecutive

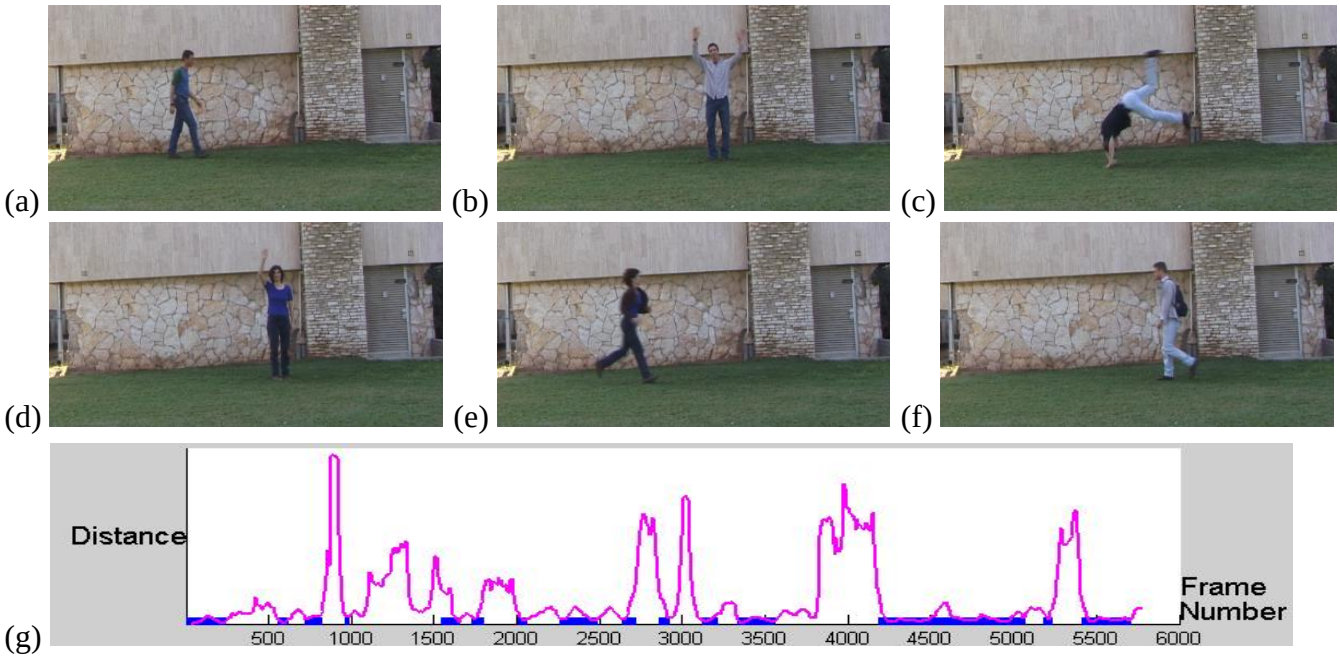

Figure 4. Event-Based Detection: *(a)-(f) Representative frames of a 6000 frame long video sequence of various people performing various activities. (g) The measured distances (using Eq. (2)) between a single example clip showing a person walking and a sliding window shifted across the entire sequence. The blue bars mark ground-truth for walks (manually detected and marked). The measured distance between the example clip and all other "walks" is small, even though the various "walks" were performed by different people wearing different clothes, viewed from different viewing angles, from different distances. For color images and sequences with results see* **http://www.wisdom.weizmann.ac.il/∼vision/EventDetection.**

sub-sequences we estimate the distance between them based on information from all temporal scales using the distance measure of Eq. (2). This results in a set of distance values where maxima points correspond to start/end points of events (See Figure 6).

Figure 7 displays the results of applying temporal segmentation to the sequence of Figure 5. A single action in this sequence (i.e., a single punch, kick or duck) usually extends between 25 and 40 frames. We tested our temporal segmentation scheme using a sliding temporal window of length $T = 32$ (top row of Figure 7). This resulted in good temporal segmentation (i.e., the event-based cuts were indeed detected at transitions between different actions). Small variations in the length T of the temporal window did not affect the segmentation results. However, using significantly smaller sliding temporal windows, which capture only small fragments of the actions or the short pauses between the actions (e.g., a sliding temporal window of length $T = 16$ or $T = 8$ frames) resulted in inferior temporal segmentation (middle and bottom rows of Figure 7). When using $T = 16$ all correct cuts were detected, but additional (false) temporal cuts were also detected. In most cases, these occurred at the short pauses between repeated occurrences of the same action. However, a very small temporal window (e.g., of length $T = 8$) captures too little information about the action. Hence, when a window of length $T = 8$ was used, some of the correct cuts were missed, while other false cuts were detected.

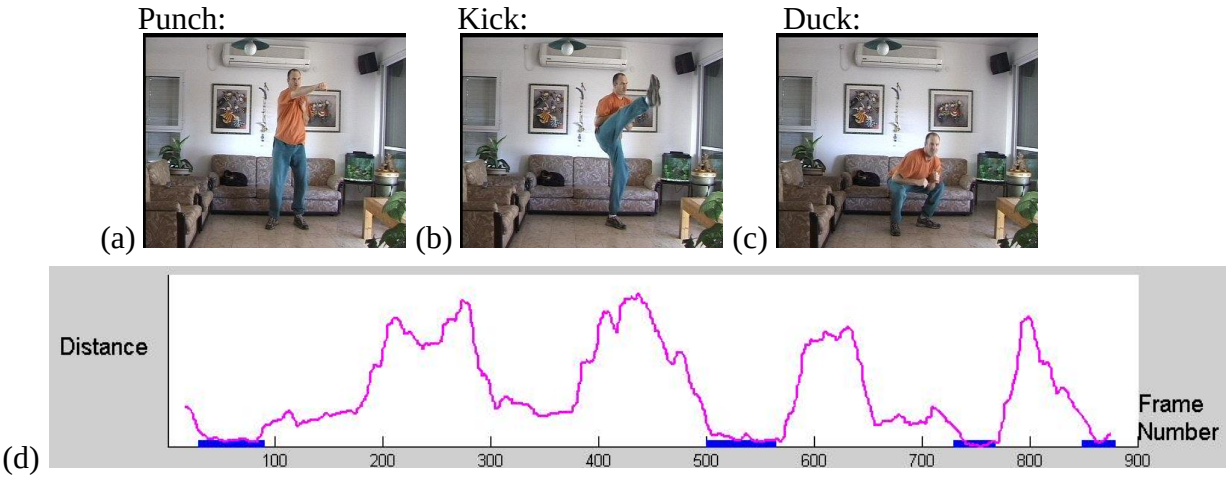

Figure 5. Event-Based Detection: *(a)-(c) Sample frames from a sequence showing a person punching, kicking and ducking. (d) The measured distances (using Eq. (2)) between a single punch clip and a sliding window shifted across the entire sequence. The blue bars mark ground-truth for punches (manually detected and marked). The measured distance between the example clip and all other "punch" clips is small. For sequences with results see* **http://www.wisdom.weizmann.ac.il/∼vision/EventDetection.**

6 Event-Based Clustering

The temporal segmentation scheme described above can be performed on-line as video streams in. However, when the entire video sequence is available (e.g., in a batch mode) and when the sequence contains multiple occurrences of events (e.g., the same events performed by different people at different times), then our event-based distance measure can further be used for clustering events within long continuous video sequences into event-consistent clusters. This can be done without prior knowledge of the types of events, their models or their temporal extents.

Here too, we use a sliding temporal window to compare sub-sequences of the long video sequence, but now every sub-sequence of length T is compared against *all* other sub-sequences of the same length within the long video sequence (and not only against its consecutive sub-sequence, as in the temporal segmentation of Section 5). Figure 8) illustrates this. We first construct a distance matrix $\mathcal{D}$ whose entries are $\mathcal{D}(i,j) = \mathcal{D}(j,i) = D_{ij}^2$, where D_{ij}^2 is the distance between sub-sequence i and sub-sequence j computed using Eq. (2). We then use the *Modified-Ncut* approach of [24, 12] to cluster the data. We start by estimating the affinity (similarity) matrix M whose entries are $\mathcal{M}(i,j) = \mathcal{M}(j,i) = \exp\left[-\mathcal{D}(i,j)^2/\sigma\right]$, where σ is a constant scale factor used for stretching values (see [26] for more details). We take multiple eigen-vectors of $\mathcal{M}$ (we chose the number of eigen-vectors to be equal to the number of clusters) and then perform k-means clustering on the entries of these vectors (for further details see [12]). This results in a fully automatic clustering process where the only manually defined parameter is the number of clusters. The initial clustering is then refined by re-classifying all sub-sequences using cluster representatives (see Section 7).

Figure 9.a displays the results of applying the above clustering method to the "punch-kick-duck" sequence of Figure 5. Results are shown again for sliding temporal windows of sizes $T = 32$ frames (which is approximately the length of a complete action), $T = 16$ frames, and $T = 8$ frames. The

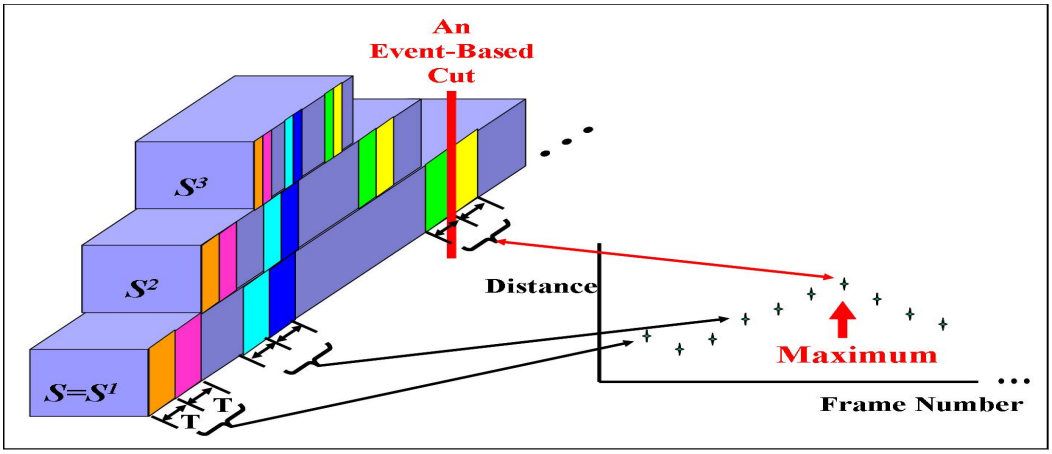

Figure 6. Temporal Segmentation Scheme: *We measure the distance between consecutive sub-sequences of length T (using all temporal scales). Maxima points correspond to temporal cuts.*

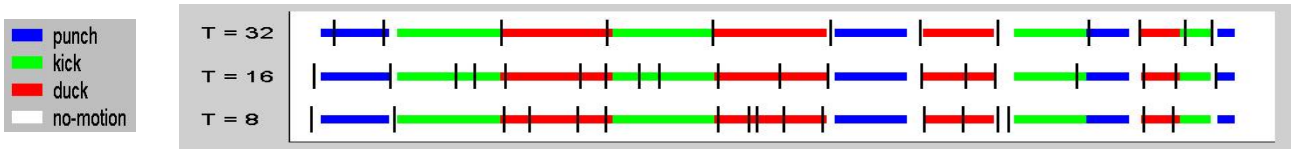

Figure 7. Temporal Segmentation: *Results of temporal segmentation for the sequence of Figure 5 using temporal windows of lengths $T = 32$ frames (top row), $T = 16$ frames (middle row) and $T = 8$ frames (bottom row). The detected cuts are marked by black vertical lines on top of the ground-truth values (manually marked by colored bars). The blue bars correspond to sub-sequences of "punches", the green bars correspond to "kicks" and the red bars correspond to "ducks". The white regions correspond to sub-sequences in which no-motion occurred. For more details see text (Section 5). For color images and sequences with results see* **http://www.wisdom.weizmann.ac.il/∼vision/EventDetection.**

number of clusters was set to three. We visually display the clustering results by color coding the time axis with the respective cluster association ($T = 32$ top colored-bar, $T = 16$ second colored-bar, and $T = 8$ third colored-bar of Figure9.a). These results can be compared against the ground-truth manual classification (bottom colored-bar of Figure9.a). For $T = 32$ all the sub-sequences were clustered correctly, and good temporal segmentation into event-consistent sub-sequences (of various lengths) was obtained. The use of $T = 16$ also yielded high quality results (almost all sub-sequences were clustered correctly and only a few were miss-classified). However, when $T = 8$ was used the results were not as good (there were more miss-classifications). The majority of the miss-classified sub-sequences correspond to short pauses where hardly any motion occurred. The clustering process classified these to the most similar cluster, which was not always the correct answer. Using a larger window (e.g., $T = 32$, or $T = 16$) we did not encounter this problem, because sub-sequences which included short pauses also included enough activity information in order to be classified correctly.

The suggested clustering process groups all the sub-sequences of length T into event-consistent clus-

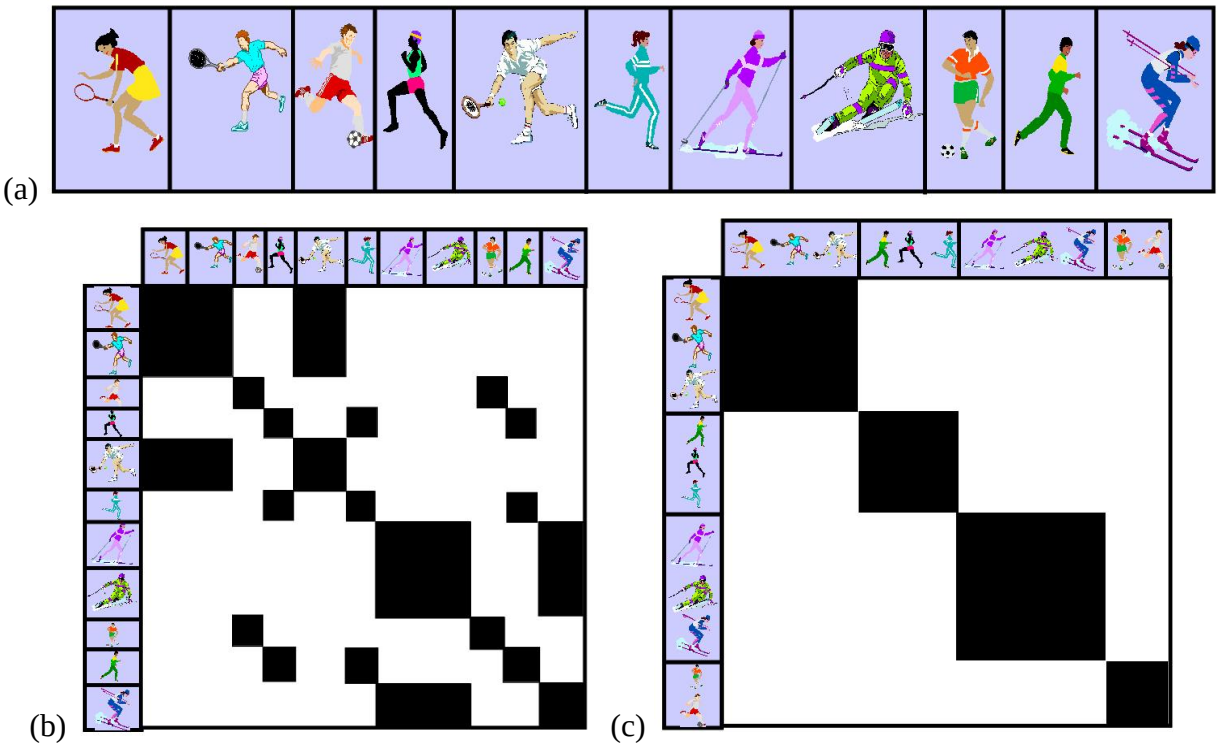

Figure 8. An illustration of the clustering scheme: *(a) A schematic drawing of the content of a video sequence showing different activities performed by different people. (b) Estimating the distances between every sub-sequence of length T to all other sub-sequences of the same length should give a distance matrix with low values (marked in black) for sub-sequence showing the same event and high values (marked in white) for sub-sequence showing different events. (c) The sorted (block diagonal) distance matrix after clustering. All sub-sequences showing the same event are grouped together.*

ters. As a by-product, this grouping also provides an event-based temporal segmentation of the long video sequence. Due to the increase in the amount of available information, the clustering process is less sensitive to the length T of the examined temporal windows, leading to a more robust temporal segmentation than that obtained by the on-line process described in Section 5. Comparing the temporal segmentation results obtained by the on-line process of Section 5 (Figure 7) with those obtained by the batch clustering process (applied to the "punch-kick-duck" sequence shows that using the "correct" window size of $T = 32$ gave high quality results in both cases (see Figure 9). Using a "wrong" window size of length $T = 16$ still gave high quality results when applying the clustering process, whereas inferior results were obtained with the on-line the temporal segmentation scheme. Using a very small temporal window of size $T = 8$ gave results of lesser quality both when applying the on-line temporal segmentation scheme and when applying the off-line (batch) event-based clustering.

To further illustrate the robustness of the clustering process to inaccuracies in specified parameters, we show in Figure 10 results of applying the same clustering method to the "punch-kick-duck" sequence of Figure 5 but this time with varying (wrong) number of clusters (instead of the correct number of 3).

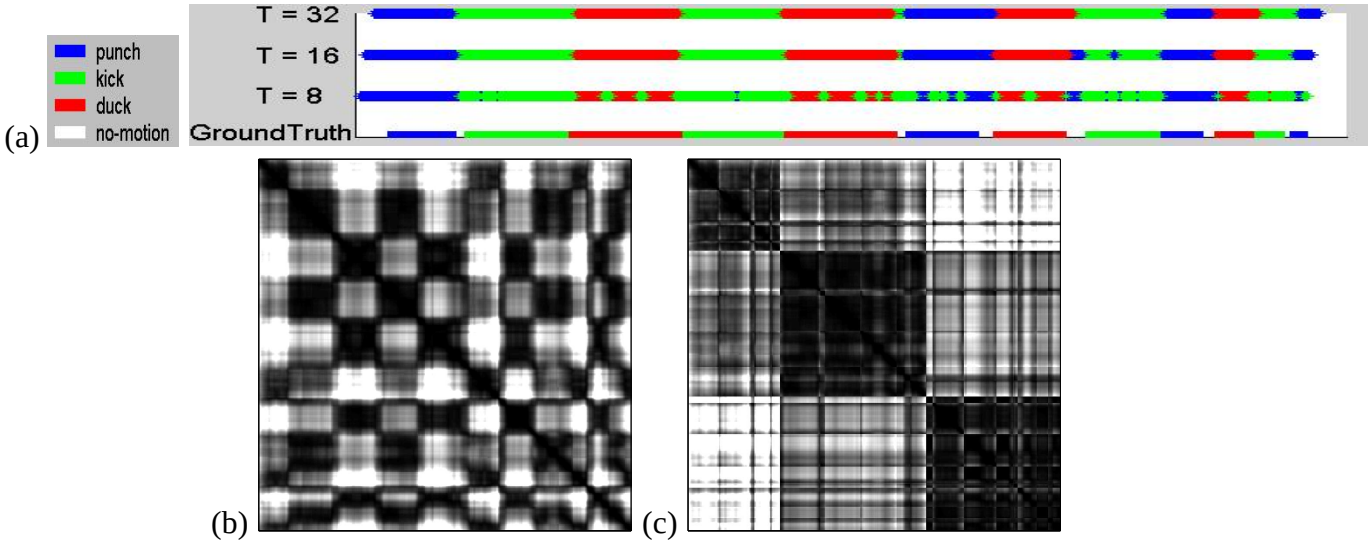

Figure 9. Event-Based Clustering: *(a) Clustering results for the sequence of Figure 5 (using 3 clusters; see Section 6) displayed on the time axis for $T = 32$ (top colored bar), $T = 16$ (second colored bar) and $T = 8$ (third colored bar) vs. ground-truth information (bottom colored bar). All sub-sequences corresponding to the same cluster were assigned the same color (blue for the "punches" cluster, green for the "kicks" cluster, and red for the "ducks" cluster). (b) The distance matrix before clustering. (c) The distance matrix after clustering.*

When the number of clusters is set to 2 (Figure 10.a) we get that all "punches" and all "kicks" are grouped together into one cluster and the second cluster corresponds to all the "ducks". This result is intuitive, since the "punches" and "kicks" are far more similar to each other than to the "ducks". Similar results were obtained for $T = 32$, $T = 16$ and for $T = 8$. Setting the number of clusters to the wrong number 4 (Figure 10.b) forced the clustering process to classify some of the sub-sequences into a fourth cluster (marked in black). When $T = 32$ and $T = 16$ were used, the forth cluster corresponded to a sub-cluster of the "punches". One sub-cluster contained all the sub-sequences in which the punching person stood frontal to the camera, and the other sub-cluster of the "punches" included all the punch sub-sequences with a small angle between the person and the camera (However, for $T = 16$ ther were more miss-classifications). When $T = 8$ was used, the fourth cluster corresponded mostly to the short pauses between consecutive punches or consecutive kicks or consecutive ducks, where no activity was performed. Since such pauses last for only extremely short periods of time, they cannot be detected when a temporal window of length $T = 32$ is used.

Figure 11 shows the results of applying the event-based clustering method to the long video sequence (6000 frames) of Figure 4. We used a sliding temporal window of size $T = 64$ frames, at skips of 8 frames (this is done to reduce the dimensions of the distance matrix from 6000×6000 to 750×750 in order to speed-up the clustering process). As described before, the sequence contains four types of frequently occurring activities: walking, jogging, hand-waving, and walking-in-place (performed by different people of both genders wearing different clothes for different lengths of time), and single occurrences of several other activities (e.g., rolling, and other free activities). Most of the walking is performed parallel to the image plane, but several parts include walking in slightly diagonal directions

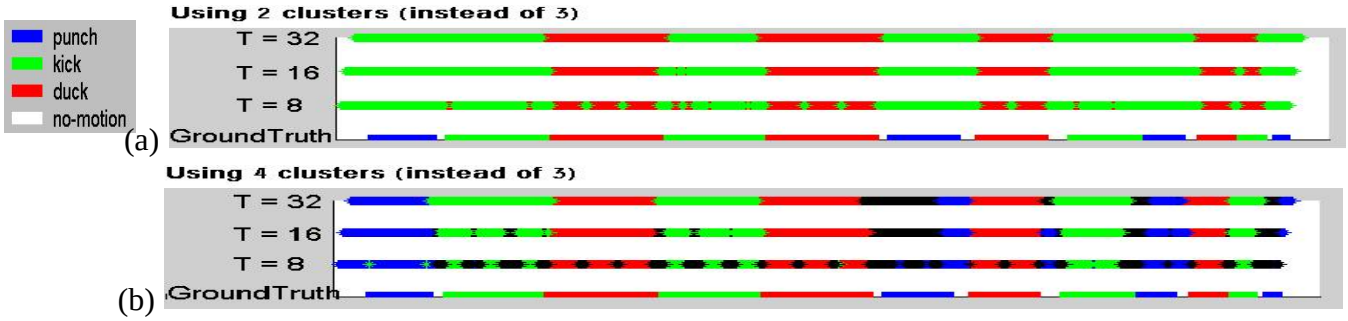

Figure 10. Robustness of Event-Based Clustering: *Clustering results using "wrong" number of clusters and various temporal windows. (a) Clustering results for the sequence of Figure 5 when the number of clusters is set to 2 (instead of 3). Results are shown for temporal windows of length $T = 32$ frames (top row), $T = 16$ frames (second row) and $T = 8$ frames (third row). (b) Clustering results when the number of clusters is set to 4 (i.e., over-fragmentation). See text for more details.*

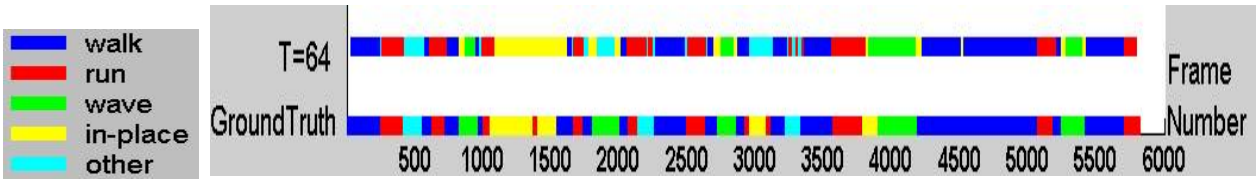

Figure 11. Event-Based Clustering: *Results of event-based clustering for the sequence of Figure 4 using a temporal window of length $T = 64$ frames. Top row: All sub-sequences corresponding to the same cluster were assigned the same color (blue for the "walks" cluster, red for the "runs" cluster, etc.). Bottom row: manually marked ground-truth information. See text for more details. For color images and sequences with clustering results see* **http://www.wisdom.weizmann.ac.il/∼vision/EventDetection.**

and some on snake-like paths. Waving includes waving with a single hand or both hands (not necessarily having the same phase). Figure 11) shows the clustering result vs. ground-truth.

Figure 12 shows the result of applying event-based clustering to a 500-frame long tennis sequence recorded with a panning camera. The sequence was first stabilized to compensate for the camera-induced background motion using [9]. A sliding window of size $T = 10$ was applied to the stabilized sequence. The three detected clusters correspond to *strokes* (backhand and forehand), *hops*, and *steps* of the tennis player. Since our normalized local measurements are invariant to mirror reflections of the same action, the backhand and forehand strokes are clustered together into a single "strokes" class. Figure 12.d shows the clustering results vs. ground-truth.

7 Refining the Representation and Measure

When only a single example clip of each event E is available, the event representation (i.e., the empirical distributions of components of space-time measurements at multiple temporal scales) is constructed from the single example clip, and the distance between two event clips is estimated using the χ^2 test (see Eq. (2)).

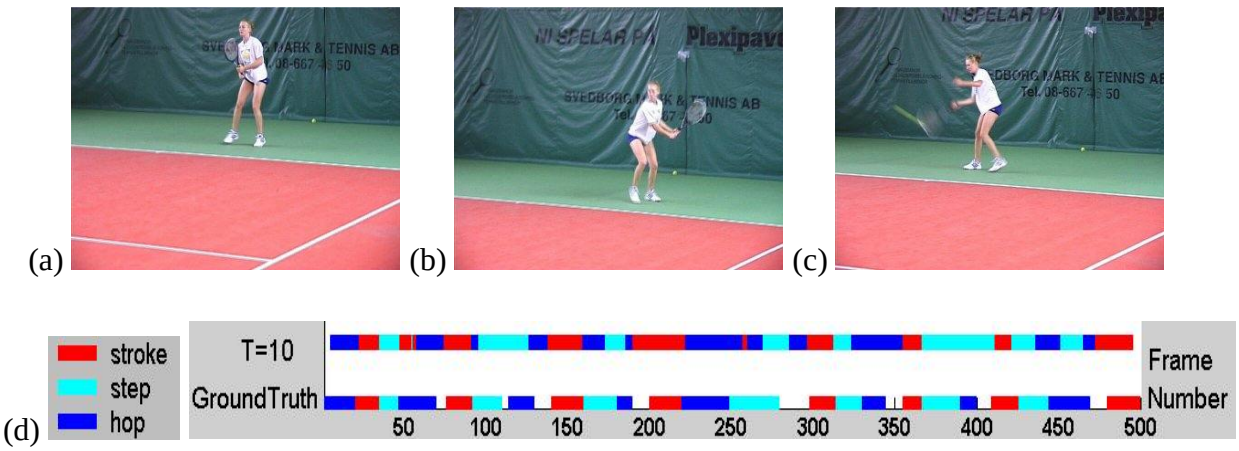

Figure 12. Event-Based Clustering: *(a)-(c) Representative frames of a 500 frame long video sequence. (d) Clustering results displayed on the time axis (top colored-bar) vs. ground-truth information (bottom colored-bar). See text for more details. For color images and sequence with clustering results see* **http://www.wisdom.weizmann.ac.il/∼vision/EventDetection.**

However, when multiple example clips of the same event E are available (either specified manually, or obtained via the automatic clustering process), we can refine the event representation and the distance measure to emphasize the contribution of important space-time measurements at the important temporal scales, as learned from the examples.

To show this, we first rewrite the χ^2 distance measure as a *weighted sum* of the squared differences between two empirical distributions h_1 and h_2:

$$\chi^2 = \sum_i \frac{(h_2(i) - h_1(i))^2}{h_2(i) + h_1(i)} = \sum_i w_i (h_2(i) - h_1(i))^2 \tag{3}$$

where $w_i = \frac{1}{h_2(i)+h_1(i)}$. Treating each histogram h as a column vector, we can re-write this in vector notation:

$$\chi^2 = (h_2 - h_1)^T \left[diag(h_2 + h_1)\right]^{-1} (h_2 - h_1) \tag{4}$$

where $diag(h_2 + h_1)$ is a diagonal matrix whose i-th diagonal entry is $h_2(i) + h_1(i)$.

We next show how to refine the weights w_i and the empirical distributions h with the gradual increase in information about the underlying data.

When multiple example clips of the same event type E are available, we compute the *mean* and *variance* of all the corresponding distributions (separately for each histogram bin of each filter response at each temporal scale). The mean histogram $\overline{h_E}$ can be used as the event representation, and the histogram of variances var_E indicates the reliability and hence the relative significance of the individual histogram bins. This is illustrated in Figure 13. The solid blue line corresponds to a mean histogram $\overline{h_E}$, whereas the dashed green lines define the envelope corresponding to the mean $+/-$ the standard deviation of all histograms of the example clips. Therefore, when estimating the distance measure between the event E (represented by $\overline{h_E}$) and any new incoming sequence with an empirical distribution h, the weights of Eq. (3) should be replaced with $w_i = \frac{1}{var_E(i)}$ (where $var_E(i)$ is the variance of all the example-histograms at bin i). Namely, high weights are assigned to bins of low variance (which are more reliable, e.g., near the dashed cyan arrow) and low weights to bins of high variance (which are

less reliable, e.g., near the solid magenta arrow). The refined distance measure specialized for detecting events similar to E is therefore:

$$D_E^2 = \left(h - \overline{h_E}\right)^T [diag(var_E)]^{-1} \left(h - \overline{h_E}\right) \quad (5)$$

This measure identifies and emphasizes the contribution of prominent spatio-temporal components of the space-time measurements at their prominent temporal scales. Note that for each event type E there will be a different set of weights.

When neighboring histogram bins (which correspond to similar filter responses) are not statistically independent, we can further generalize the distance measure of Eq. (5) by incorporating covariance information and not only the variance:

$$D_E^2 = \left(h - \overline{h_E}\right)^T {cov_E}^{-1} \left(h - \overline{h_E}\right) \quad (6)$$

This is actually the squared mahalanobis distance [6], applied here to distributions (histograms).

Figure 14 compares the quality of the χ^2-based distance measure of Eq. (2) (or Eq. (3)) to the refined distance measure of Eq. (5) for detection purposes. The detection is based on the measured distance between a single (64 frames long) example clip of a walking event compared against a sliding window (of 64 frames) which was shifted across a few-minute-long (6000 frames long) video sequence (the sequence of Figure 4). The bottom colored-bar marks the ground-truth (manually detected walks). The top colored-bar shows the results using the distance measure of Eq. (2). Even though the example clip contains only a single person walking in a single direction and wearing a particular set of clothes, all the other walking people wearing different clothes and walking in different directions were detected. Although there are several false detections, the result indicates that our initial choice of representation and distance measure are reasonable given no other information. The middle colored-bar shows the detection results using the refined distance measure of Eq. (5) based on 10 example clips of walking events (each 64 frames long). Using the refined distance measure reduces the number of false detections.

7.1 A Bayesian Point of View

The problem of event detection can be reposed as follows: Given a new video clip S and an event type E, what is the a posteriori probability $P(E|S)$?

According to Bayes rule $P(E|S) = \frac{P(S|E)P(E)}{P(S)}$. When no information is available about the set of possible events, the number of events, or their frequency of occurrence, we assume that all events E are equally likely (i.e., $P(E)$ is the same of all E). Similarly, when no information is available about the types of sequences, we assume that all sequences S are equally likely (i.e., $P(S)$ is the same of all S). In that case $P(E)/P(S)$ is constant, and

$$P(E|S) \propto P(S|E) = \frac{1}{(2\pi)^{r/2} \mid Cov_E \mid^{\frac{1}{2}}} exp\left[-\frac{1}{2}\left(h - \overline{h_E}\right)^T cov_E^{-1}\left(h - \overline{h_E}\right)\right],$$

where h denotes the empirical distribution of space-time measurements in the video clip S, and r is the dimension of h (i.e., the number of bins). The log-likelihood of E is proportional to the (negated) squared mahalanobis distance defined in Eq. (6). Therefore, small distances measured by Eq. (6) directly correspond to high likelihood of the event E, and vice versa. When there is an approximate knowledge about the size of the constant $\frac{P(E)}{P(S)}$, this can be used to determine the choice of threshold to be applied to the distance measure of Eq. (6) for the purpose of event detection and event-based indexing. If we

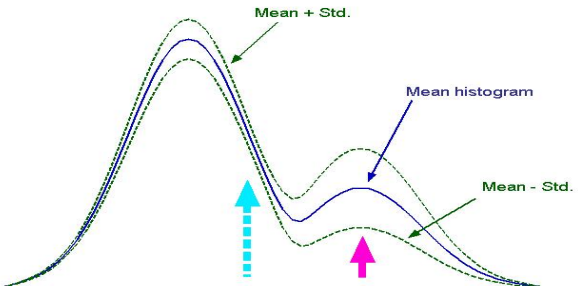

Figure 13. Weighting Histogram Bins: *The mean histogram $\overline{h_E}$ of all the distributions (histograms) of the example clips (blue solid line) and the standard deviation from it (green dashed lines) of all these distributions. The dashed cyan arrow points at a low variance region and the solid magenta arrow points at a high variance region.*

further assume independence between histogram bins, then the off-diagonal entries of the covariance matrix become zero, and the discussion above applies then to the distance measure of Eq. (5). When there is no prior statistical information about the event E (e.g., when there is only one example clip of event type E), then our probability estimate is based on the standard χ^2 distance measure of Eq.(4) (which is equivalent to Eq. (2) or Eq. (3)).

Let us now examine the case when multiple example clips are available for all event types (e.g., via the clustering process). We can further use our improved distance measures to refine the clustering results, as well as for classification of new incoming sequences. This is done as follows: Let $E_1, ..., E_N$ be the set of all possible events, and let S be a sequence to be classified. Then $P(E_k|S) = \frac{P(S|E_k)P(E_k)}{P(S)} = \frac{P(S|E_k)P(E_k)}{\sum_n P(S|E_n)P(E_n)}$. The priors $P(S|E_1), ..., P(S|E_N)$ can be estimated from the distance measure (the initial or refined, depending on the number or example clips per event type), as explained above. When all events are assumed to be equally likely (i.e., $\forall k \;\; P(E_k) = \frac{1}{N}$), then $P(E_k|S) = \frac{P(S|E_k)}{\sum_n P(S|E_n)}$. Alternatively, one can assume that the prior $P(E_k)$ of an event E_k is proportional to the frequency of its occurrences. This can be estimated by the number of times it was detected in a long video sequence, or by the size of each cluster identified in the clustering process. A sequence S will be classified as event type E_k if $P(E_k|S) = \max_n(P(E_n|S))$. The clustering results shown in Figs. 11 and 12 were obtained by simple clustering followed by classification refinement according to representative event types.

8 Conclusion

We proposed a simple statistical distance measure between video sequences based on their behavioral content. This measure is non-parametric, and can thus handle a wide range of dynamic events without prior knowledge of the types of events, their models, or their temporal extent. We showed that this measure can be used for a variety of video applications, including event-based detection, indexing, temporal segmentation, and clustering of long streams of video sequences.

While our event-based distance measure is inferior in accuracy to the more sophisticated (but more restricted) parametric models, it can be refined with the gradual increase in available information about the underlying data.

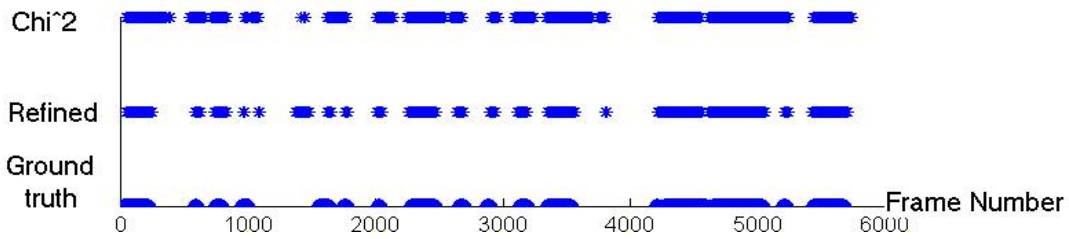

Figure 14. Refined Event Detection: *The results of detecting a walking event within the long (6000 frames) video sequence of Figure 4. Top colored bar: results using the distance measure of Eq. (2). Middle colored bar: the results using the refined distance measure of Eq.(5). Bottom colored bar: ground-truth (manually detected and marked). See text (Section 7) for more details.*

References

[1] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics*, 2001. to appear.

[2] M. J. Black and Y. Yacoob. Recogniziong facial expressions in image sequences using local parametrized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[3] J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *SIGGRAPH*, pages 361–368, 1997.

[4] O. Chomat and J. L. Crowley. Probabilistic recognition of activity using local appearnce. In *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, USA, June 1999.

[5] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):781–796, August 2000.

[6] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classifi cation.* John Wiley and Sons, Inc., New York, 2001.

[7] D.M. Gavrila and L.S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, SF, 1996.

[8] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238, 1997.

[9] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16, February 1994.

[10] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parametrized model of articulated image motion. In *2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 38–44, Killington,Vermont, Oct. 1996.

[11] F. Liu and R. W. Picard. Finding periodicity in space and time. In *International Conference on Computer Vision*, Bombay, India, January 1998.

[12] M. Meila and J. Shi. A random walks view of spectral segmentation. In *International Workshop on Artifi cial Intelligence and Statistics*, Key West, Florida, January 2001.

[13] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Databas Systems II, IFIP*, 1992.

[14] C.W. Ngo, T.C. Pong, H. Zhang, and R.T. Chin. Detection of gradual transitions through temporal slices analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 36–41, June 1999.

[15] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking fi gures in xyt. In *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994.

[16] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. In *Workshop on Non-Rigid Motion and Articulated Objects*, Austin, Texas, November 1994.

[17] R. Polana and R. Nelson. Recognition of motion from temporal texture. In *IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, June 1992.

[18] R. Polana and R. Nelson. Detecting activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, New-York, June 1993.

[19] Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.

[20] D.F. Shu S. Swanberg and R. Jain. Knowledge guided parsing in video databases. In *SPIE*, 1993.

[21] P. Saisan, G. Doretto, and S. Soatto Y.N. Wu. Dynamic texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, December 2001.

[22] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive fi eld histograms. In *International Conference on Pattern Recognition*, Vienna, Austria, August 1996.

[23] A. Schodl, R. Szelsiki, D.H. Salesin, and I. Essa. Video textures. In *SIGGRAPH*, 2000.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.

[25] M. Szummer and R. W. Picard. Temporal texture modeling. In *Int. Conf. on Image Processing*, Lausanne, Sep. 1996.

[26] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 1999.

[27] Y. Yacoob and M. J. Black. Parametrized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[28] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 123–130, Hawaii, December 2001.

[29] H. Zhang, A. Kankanhali, and W. Smoliar. Automatic partitioning of full-motion video. In *Multimedia Systems*, 1993.