Hierarchical Spatio-Temporal Context Modeling for Action Recognition

Ju Sun[1,3], Xiao Wu[2], Shuicheng Yan[3], Loong-Fah Cheong[3], Tat-Seng Chua[4], Jintao Li[2]

[1]Interactive and Digital Media Institute, National University of Singapore, Singapore

[2]Institute of Computing Technology, Chinese Academy of Sciences, P. R. China

[3]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

[4]School of Computing, National University of Singapore, Singapore

Abstract

The problem of recognizing actions in realistic videos is challenging yet absorbing owing to its great potentials in many practical applications. Most previous research is limited due to the use of simplified action databases under controlled environments or focus on excessively localized features without sufficiently encapsulating the spatio-temporal context. In this paper, we propose to model the spatio-temporal context information in a hierarchical way, where three levels of context are exploited in ascending order of abstraction: 1) point-level context (SIFT average descriptor), 2) intra-trajectory context (trajectory transition descriptor), and 3) inter-trajectory context (trajectory proximity descriptor). To obtain efficient and compact representations for the latter two levels, we encode the spatio-temporal context information into the transition matrix of a Markov process, and then extract its stationary distribution as the final context descriptor. Building on the multi-channel nonlinear SVMs, we validate this proposed hierarchical framework on the realistic action (HOHA) and event (LSCOM) recognition databases, and achieve 27% and 66% relative performance improvements over the state-of-the-art results, respectively. We further propose to employ the Multiple Kernel Learning (MKL) technique to prune the kernels towards speedup in algorithm evaluation.

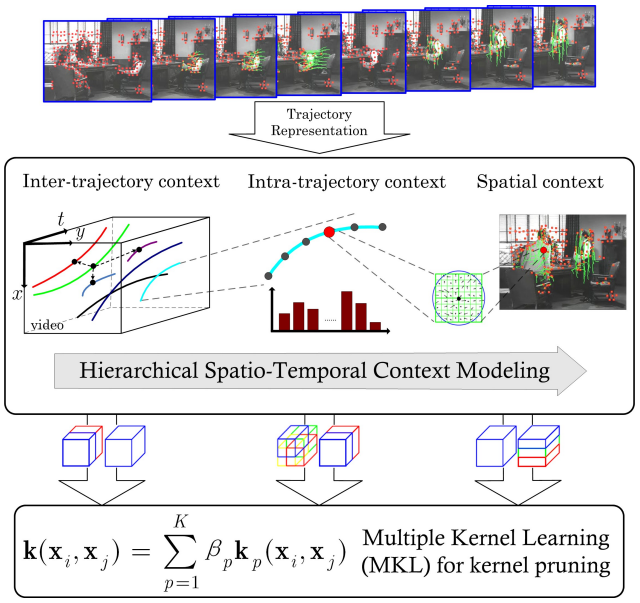

Figure 1. **Schematic diagram on hierarchical spatio-temporal context modeling.** The three levels of spatio-temporal context residing with SIFT-based trajectories are 1) the point-level context (SIFT average descriptor), 2) intra-trajectory context (trajectory transition descriptor), and 3) inter-trajectory context (trajectory proximity descriptor). They can be fed into the multiple-kernel learning module for kernel pruning to speed up subsequent processing. (Refer to the electronic version for best view)

1. Introduction

Recently, recognizing actions in unconstrained videos has been an active research topic in computer vision. There exist two sources of impetus to this topic: 1) The continual advances in high-level vision research, *e.g.* object detection and recognition. These problems share many similarities as well as difficulties with action recognition. Hence successful techniques of solving high-level vision problems can often be adapted to action recognition; and 2) The great potentials in practical applications, *e.g.* realtime video surveillance and security monitoring, automatic video indexing, and human-computer interfaces.

The problem of recognizing actions in videos is challenging, and all difficulties associated with object detection and recognition task, such as large intra-class variations, partial occlusions, low resolution and cluttered background, may also be encountered in action recognition problem. Therefore the action recognition algorithms [22, 23], which aimed to achieve the robustness to viewpoint change based on geometric reconstruction, are doomed due to their reliance on exact knowledge of the object contours or multiple view geometries, which are prone to errors in unconstrained videos. In contrast, the *bag of words* related algo-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

rithms [6, 12, 20] handled the above difficulties well and achieved promising results on some action video databases. However, this line of research tends to focus on designing sophisticated learning models, but is limited on the use and effective modeling of action-related features, in particular spatio-temporal context of the visual features. Most previous algorithms for action recognition were evaluated on video databases under controlled settings, such as the KTH [14] and figure skating [21] databases with static background or dominant single-body motion. These databases are far less complicated than the realistic movie or news video actions, *e.g*. the recently released HOHA [6] and LSCOM [20] databases.

The term *context* is widely used in both computer vision and multimedia areas, but hitherto has been loosely defined due to its task-specific nature (see [11, 19]). For the action recognition problem defined in a 3D spatio-temporal space, context refers to *any spatio-temporal information that encapsulates the spatio-temporal layout and transition, relative position, global and semi-local statistics,* etc. *of the low level visual features, e.g. gray-levels, gradients, and colors*. It is evident that context information is very important for action recognition in unconstrained videos owing to the capability to express the dynamic and structural nature of motions.

In this work, we model the spatio-temporal context information encoded in unconstrained videos based on the SIFT-based [10] trajectory, in a hierarchy of three abstraction levels (see Fig. 1): at the fine level, localized statistics of spatial gradients (SIFT average description) along the trajectory; at the intermediate level, the transition and dynamics of the trajectory in spatio-temporal domain; and at the coarse level, the spatio-temporal co-occurrence and distribution of the trajectories. By encoding the latter two levels of context information into the transition matrix of Markov process, we can derive compact and efficient representations of context based on their stationary distributions. Employing the multi-channel nonlinear SVMs as in [6] to fuse feature channels, we validate our proposed framework over two realistic action databases used in [6, 20], and demonstrate its superiority over the state-of-the-art. To reduce the cost of greedy search for best combination of channels in multi-channel SVMs, we propose to apply MKL technique [17] first to select the candidate channels before performing exhaustive search.

2. Related Work

Trajectory-based action recognition has been extensively studied in the past few years. These proposed algorithms typically differ on how to encode the dynamics of trajectories for subsequent processing. By using geometric properties such as the trajectory speed and location [4, 16], trajectory curvature [13] and trajectory segments [2], together with other cues such as local appearance [4, 16], spatio-temporal saliency and shape structure [5], many approaches have partially succeeded in modeling trajectories and capturing the context information. In this work, our main target is to describe the spatio-temporal context based on SIFT-trajectory, rather than those less structured spatially or spatio-temporally salient points. By comparison, the trajectory-based context has closer relation to motions, which are the elements for action recognition. Hence we propose a hierarchical framework to model the spatio-temporal context in three levels, namely point-level context, intra-trajectory context, and inter-trajectory context. The three level of context provide a hierarchy of multiresolution information on variation and dynamics of visual patterns inside video sequences, and hence offer more discriminative representation for characterizing different types of actions in unconstrained videos.

Compared to those approaches that build models on exact knowledge of trajectory such as the speed or even acceleration (curvatures) [4, 13, 16], our model is less sensitive to noise and does not require preprocessing such as smoothing. Different from the synthesized trajectory segments (or star diagram [2]) method which has collected displacement vectors together and analyze the distribution pattern, our approach explicitly accounts for the sequential order of the trajectory segments by finite-state Markov chain, and maximally encapsulates the dynamic nature of the trajectories.

3. Hierarchical Spatio-temporal Context Modeling in Realistic Action Videos

We propose to capture visual motion patterns by extracting the trajectories of the salient points, and then model the spatio-temporal context information residing with these trajectories. Given a video sequence $\mathbf{V}$ in $xyt-$space, we first extract a bunch of trajectories $\{\mathbf{T}_1, \cdots, \mathbf{T}_N\}$ as spatio-temporal curves, and then form three levels of context representations $\{\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3\}$ based on these curves, where $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$ are designed to describe the point-level, intra-trajectory level, and inter-trajectory level context, respectively. We start with extracting the trajectories of the spatially salient points.

3.1. SIFT-based Trajectory Extraction

Reliable spatially salient point detection and tracking algorithms are critical for the modeling of motion patterns in realistic videos. For spatially salient point detection and representation, we adopt the well established SIFT [10] technique, the effectiveness of which has been validated in numerous visual tasks, such as object recognition [9] and robot navigation [15]. The renowned robustness and scale-invariance properties of SIFT render it a better choice as compared to other techniques such as the Harris and

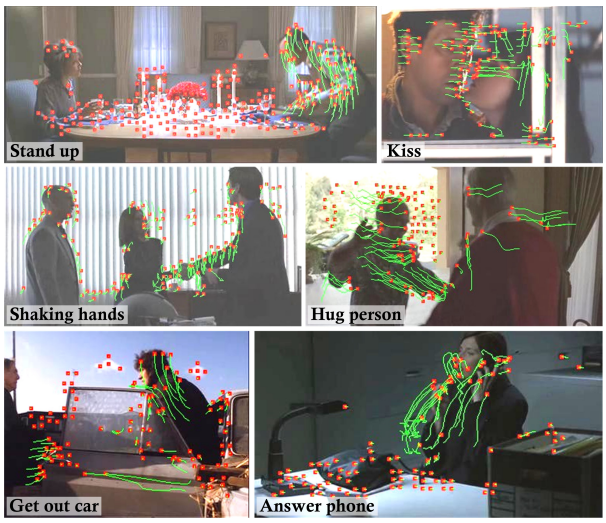

Figure 2. **Example trajectories of the SIFT salient points for six types of actions.** Note that different videos may have different frame dimensions. A pictorial trajectory presented in the figure is formed by connecting all the spatially salient points on the same trajectory. (Refer to the electronic version for best view)

Kanade-Lucas-Tomasi (KLT) feature trackers [18]

The trajectory extraction process is based on the pairwise SIFT matching over consecutive frames. For the frames $\{\mathbf{f}_1, \cdots, \mathbf{f}_k\}$ of a video sequence denoted $\mathbf{V}$ with k frames, we establish all the SIFT point matches between $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$, for $1 \leq i \leq k-1$. Matches that extend over several frames then form a motion trajectory of the SIFT salient point. To mitigate the effect of incorrect matches and hence the creation of spurious trajectories, we impose the unique-match constraint and also discard matches that are too far apart, since most realistic motions cannot be very fast. To be specific, for any SIFT salient point p in frame i, there can be maximally one candidate match point p' in frame $i+1$, and p' must be located within a $N \times N$ (we set $N = 64$ in all the experiments) spatial window around point p. This windowing approach ensures that the trajectory may automatically end when reaching the shot boundaries or with considerable occlusions. In such situations, our tracking algorithm will restart to track another batch of trajectories. To further remove possible noisy trajectories and reduce the chance of long trajectories mixing up with successive motions, *e.g.* human stand up and walk, we restrict the length L of any valid trajectory to be $L_{\min} \leq L \leq L_{\max}$. In this work, we set $L_{\min} = 5$ and $L_{\max} = 25$, which correspond to $0.2 \sim 1$ second in duration. Fig. 2 shows some example trajectories generated by our proposed tracking method.

3.2. Point-level Context: SIFT Average Descriptor

The importance of spatial context information in action recognition has been discussed in most previous work, *e.g.* the *what* component described in [20]. It is evident for scene and object recognition tasks that the spatial context information encoded in an image frame can provide critical implication on the semantic category of the occurring objects [11, 19]. Similarly for action recognition over frame sequences, this source of information is also very useful in constraining the action category.

The point-level context information is measured as the average of all the SIFT features extracted at the salient points residing on the extracted trajectory. For a motion trajectory of length k, the SIFT average descriptor $\mathbf{S}$ is related to all SIFT descriptors $\{\mathbf{S}_1, \cdots, \mathbf{S}_k\}$ along this trajectory by

$$\mathbf{S} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{S}_i. \quad (1)$$

The underlying philosophy of the point-level context modeling is two-fold. First, the tracking process ensures that the local image patches residing on the trajectory are stable, and thus the resultant SIFT average descriptor offers a robust representation for certain aspect of visual content within the video. Second, similar to the silhouette average used in [8] for human gait recognition, the SIFT average can also encode partially the temporal context information,

To facilitate efficient representation and processing over videos, we employ the *bag of words* model. Specifically, we construct a visual vocabulary with 1000 words by K-means algorithm over the sampled SIFT average descriptors (from the training set). We then assign each SIFT average descriptor to its closest (in the sense of Euclidean distance) visual word. The histogram of visual word occurrences over a spatio-temporal volume forms the final representation for the point-level spatio-temporal context of a trajectory.

3.3. Intra-trajectory Context: Trajectory Transition Descriptor

The point-level context characterizes mainly the *what* part of the video, namely what object components appear in the videos. For action recognition, however, the dynamic properties of these object components are more essential in characterizing the actions, *e.g.* for the action of standing up or getting out car.

The Markov chain is a powerful tool for modeling the dynamic properties of a system. Its merit mainly lies at its capability in representing directed causal and probabilistic relations. The markov stationary distribution, associated with an ergodic Markov chain, offers a compact and effective representation for a dynamic system. In this work, each trajectory or the entire video is considered as a dynamic system, and we expect to extract such a compact representation to measure the spatio-temporal context within this dynamic system. Before formally demonstrating the intra-trajectory context modeling, we give a brief introduction to the related

concepts and properties on Markov chain process. Most techinical details here can be found in [1].

A Markov chain is a sequence of random variables $[X_1, X_2, X_3, \cdots]$ with the Markov property, namely given the present state, the future and past states are independent. Formally $P(X_{n+1} = x | X_n = x_n, \cdots, X_1 = x_1) = P(X_{n+1} = x | X_n = x_n)$. The possible values of X_i form a countable set $\mathbf{S}$ called the state space of the chain. When the state space is finite (assuming K states), the transition probability distribution can be represented by a matrix $\mathbf{P}_{K\times K}$, named the Markov (stochastic) transition matrix. $\mathbf{P}_{K\times K}$ has the following three properties: 1) $p_{ij} = P(X_{n+1} = j | X_n = i)$; 2) $p_{ij} \geq 0$; and 3) $\sum_{j=1}^{K} p_{ij} = 1$.

Definition 3.1 (Ergodic Finite-State Markov Chains) *A finite-state Markov chain is said to be ergodic (aperiodic irreducible) if every state is accessible from any state* [1].

Intuitively, an ergodic Markov chain can be considered as a simple connected graph in its state space. This interpretation is crucial for our later formulations.

Theorem 3.2 *Any ergodic finite-state Markov chain is associated with a unique stationary distribution (row) vector $\boldsymbol{\pi}$, such that $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$* [1].

The following theorem provides a method to approximate the vector $\boldsymbol{\pi}$.

Theorem 3.3 *1) The limit $\mathbf{A} = \lim_{n\to\infty} \mathbf{A}_n$ exists for all ergodic Markov chains, where the matrix $\mathbf{A}_n = \frac{1}{n+1}(\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^n)$. 2) Each row of $\mathbf{A}$ is the unique stationary distribution vector $\boldsymbol{\pi}$* [1].

Hence when the ergodicity condition is satisfied, we can approximate $\mathbf{A}$ by $\mathbf{A}_n$, where

$$\mathbf{A}_n = \frac{1}{n+1}(\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^n), \text{for } n < \infty. \quad (2)$$

In all our experiments, we set $n = 50$. To further reduce the approximation error when using a finite n, $\boldsymbol{\pi}$ is calculated as the column average of $\mathbf{A}_n$.

The above discussion suggests the possibility of encoding a trajectory by the Markov stationary distribution $\boldsymbol{\pi}$ if it can somehow be converted into an ergodic finite-state Markov chain. To facilitate this conversion, a finite number of states are sought and quantization naturally gets involved[1].

For two point P and P' within two consecutive frames along the same trajectory, we denote the displacement vector $\mathbf{D} = \overrightarrow{PP'}$. Note that since the temporal component

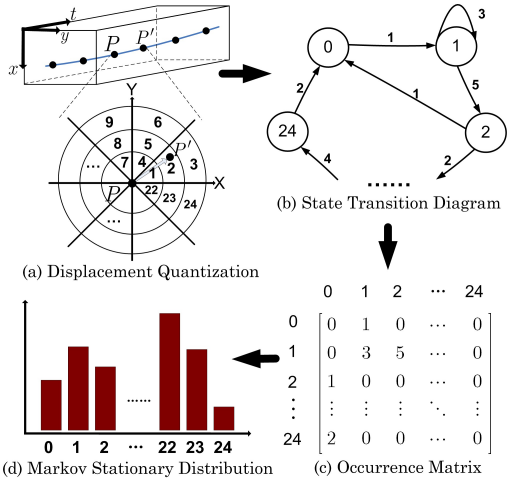

Figure 3. **Illustration of the process to generate the trajectory transition descriptor for a trajectory**. (a) Every displacement vector $\mathbf{D}$ is quantized in terms of magnitude and orientation. (a)→(b) The successive state transition is transformed into a directed graph (state-space diagram). (b)→(c) The state diagram is translated into the occurrence matrix, which is further row-normalized into a valid Markov transition matrix $\mathbf{P}$. (c)→(d) The unique stationary distribution vector $\boldsymbol{\pi}$ is computed for $\mathbf{P}$. (Refer to the electronic version for best view)

of $\mathbf{D}$ is deterministic (the frame interval), we will ignore it in subsequent discussion and assume $\mathbf{D}$ comprises spatial components only, namely $\mathbf{D} = (\Delta x, \Delta y)$. To perform a reasonable quantization on $\mathbf{D}$, we take into consideration both the magnitude and orientation (Fig. 3 (a)). For magnitude, we set 3 uniform quantization levels, whereby $\|\mathbf{D}\|$ is first normalized by the largest displacement magnitude $\|\mathbf{D}\|_{\max}$ residing with the same trajectory. This normalization is to counter the effect of scale, and thus the quantization that follows is scale invariant. For orientation, we divide the full circle into 8 equal sectors, each subtending $45°$. The combination of magnitude and orientation quantization results in 24 bins in polar coordinate. We append an additional bin to collect the zero movement, producing 25 bins. After quantizing all displacement vectors along a trajectory, we translate the sequential relations between theses vectors into a directed graph, which is similar to the state diagram of a Markov chain (Fig. 3 (b)). Here we get 25 vertices corresponding to the 25 quantization states, and weighted edges corresponding to the occurrence of each transition between the states. We further establish an equivalent matrix representation of the graph, and perform row-normalization on the matrix to arrive at a valid transition matrix $\mathbf{P}$ for a certain Markov chain (Fig. 3 (c)). Once we obtain the transition matrix $\mathbf{P}$ and make sure it is associated with an ergodic Markov chain[2], we can use Eqn. (2) to compute $\boldsymbol{\pi}$ (Fig. 3

[1]We are reluctant to employ the term Hidden Markov Model (HMM) for the current problem for avoid confusion. Compared to the three classic problems HMM deal with (see *e.g.* Sec-3.10 [3]), we have explicitly defined all states, and then generated the sparse state transition matrix. The whole procedure is feedforward and hence less evoluted, as compared to the solutions to the classic HMM problems.

[2]We initialize the state diagram with some negligible weights (we use $w_0 = 0.15$ throughout our experiments) between any two vertices before we calculate the occurrences. This trick applies to similar situations after.

(d)).

We employ the same procedure to transform every trajectory within a video into its stationary vector representation $\boldsymbol{\pi}$. In order to represent all trajectories over a spatio-temporal volume in a fixed length manner, we also employ the *bag of words* method (we set $K = 1000$) to build a histogram of trajectory occurrences based on the extracted Markov chain stationary distribution features.

3.4. Inter-trajectory Context: Trajectory Proximity Descriptor

The modeling of point-level context and intra-trajectory context provides informative clues for the actions in realistic videos. However, these context features are mainly designed for solo-actions involving only one object, and not good at characterizing the actions involving two or even more objects, *e.g.* kissing and shaking hands, *etc*. In this subsection, we introduce the inter-trajectory context modeling for offering such capabilities in characterizing action between objects.

Specific to the local features in images and videos, *bag of words* only encodes the global statistic but misses more detailed information, such as the relative position of features or local density of features, *etc*. A common way to compensate for this information loss is to use multi-scale pyramids [7] or spatio-temporal grids [6] to produce a coarse description of the feature layout, but these methods are still limited in characterizing details of the above information.

Building upon the Markov stationary distribution discussed in the previous subsection, we formulate more localized relations between trajectories in this subsection. For the sake of simplicity, we denote the spatio-temporal position of a trajectory as its geometric center $\mathbf{o}$, equivalently $\mathbf{o} = (\bar{x}, \bar{y}, \bar{t})$. Suppose there are m classes of trajectories by the K-means algorithm in the last subsection, $K_1, K_2, \cdots, K_m$. For a particular trajectory $\mathbf{s}$ of class K_i at (x_o, y_o, t_o), we consider a spatio-temporal volume $\mathbf{V}_{\mathbf{s}-xyt}$ defined as

$$\mathbf{V}_{\mathbf{s}-xyt} = \{(x, y, t) \mid |x - x_o| \leq \delta x, |y - y_o| \leq \delta y, \\ 0 \leq t - t_o \leq \delta t\}. \quad (3)$$

So $\mathbf{V}_{\mathbf{s}-xyt}$ is a cuboid spatially centered at $\mathbf{o}$ with dimension $2\delta x \times 2\delta y$ while temporally forward with a duration of δt (Fig. 4 (a)). We calculate the local occurrence statistics of all types of trajectories within the cuboid associated with $\mathbf{s}$. Repeating this process for all trajectories within the video sequences under consideration and collecting the results by trajectory classes, we arrive at an occurrence matrix $\mathbf{C}$ as,

$$c_{ij} = \sum_{\mathbf{s} \in K_i} \#\left(\mathbf{V}_{\mathbf{s}-xyt}, K_j\right), \quad (4)$$

where $\#\left(\mathbf{V}_{\mathbf{s}-xyt}, K_j\right)$ denotes the number of trajectories falling into category K_j and within the volume of $\mathbf{V}_{\mathbf{s}-xyt}$.

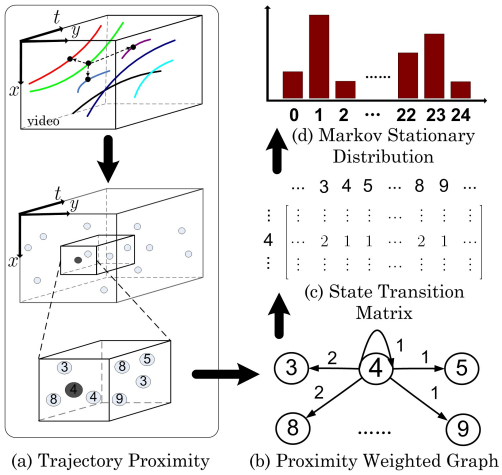

Figure 4. **Illustration of the process to generate the trajectory proximity descriptor for a spatio-temporal volume**. The spatio-temporal location of a trajectory is simplified as its geometric center. (a) For any trajectory, a local statistic of different trajectories is formed in its temporal forward proximity. (a)→(b) All such statistics within a spatio-temporal volume are summarized by a weighted directed proximity map. (b)→(c) The proximity map is converted into an occurrence matrix, and further row-normalized into a valid Markov transition matrix $\mathbf{P}$. (c)→(d) The stationary distribution vector $\boldsymbol{\pi}$ is hence obtained to represent the inter-trajectory context. (Refer to the electronic version for best view)

From this, we can convert $\mathbf{C}$ into a valid transition matrix $\mathbf{P}$ for a Markov chain process, and obtain the informative vector $\boldsymbol{\pi}$ (Fig. 4 (b) & (c)) for characterizing inter-trajectory context. The dimension of $\mathbf{V}_{s-xyt}$ is determined by the three parameters $\delta x, \delta y$ and δt. Setting these parameters properly is critical here since: 1) too small a cuboid can contain only very few or even no trajectories, and hence the distribution information provided is highly unreliable; whereas 2) a too large size cuboid encloses too many trajectories, even those that are far apart, therefore the proximity information provided is oversmoothed. In all the experiments, we set $\delta x, \delta y$ and δt to be a fraction of the video dimension in x, y, and t direction, respectively. Namely, $\delta x = x/f_x, \delta y = y/f_y$, and $\delta t = t/f_t$. This setting ensures the volume of the cuboid is adapted with the scale of the video sequences. We observe $f_x = f_y = f_t = 5$ generally gives satisfactory results.

The proposed technique here to encode the trajectory interaction information is not meant to be a replacement of the techniques such as multi-resolution and spatio-temporal grids. Rather, it aims to provide complementary local distribution information that is less structural than that conveyed by other techniques. In fact, to maximally utilize the available information, we follow [6] and apply all the 1×1, 2×2, 3×3 (not applicable to trajectory proximity descriptor), horizontal $h3 \times 1$, vertical $v3 \times 1$, and center-focused $o2 \times 2$ spatial grids, and non-overlapping equal division $t1$, $t2$, $t3$, plus centered focused $ot2$ temporal grids. The combination

of six spatial grids with four temporal binnings results in 24 possible spatio-temporal grids. For the three levels of context encoded by the SIFT average descriptor (SIFT), Trajectory Transition Descriptor (TTD), and Trajectory Proximity Descriptor (TPD), the combination of these descriptors and spatio-temporal grids brings out $\mathbf{F}_1$ ($6 \times 4 = 24$ channels), $\mathbf{F}_2$ ($6 \times 4 = 24$ channels), and $\mathbf{F}_3$ ($5 \times 4 = 20$ channels) in order, resulting in 68 feature channels in total.

4. Pruning with Multiple-Kernel Learning

These 68 feature channels essentially involve overlaps, although they are in general expected to be complementary to each other. The best performance of a particular action category generally entails only a few of the many feature channels. To optimize the combination of different feature channels and hence to produce the best prediction results pose a great challenge here. The multi-channel SVM and greedy search taken by [6] as suggested in [24] do the job in a brute-force manner, resulting in prohibitive computational requirement for large-scale problem with many feature kernels and evaluation samples. In such situation it is desirable a subset of the many channels can somehow be roughly selected, maximally retaining the channels of the candidate best combinations, followed by greedy search over over the subset of channels.

The recent development of Multiple Kernel Learning (MKL) technique provides such possibility. Different from tradition kernel-based method which allows for only one kernel, MKL considers a convex combination of K kernels [17], $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{K} \beta_p \mathbf{k}_p(\mathbf{x}_i, \mathbf{x}_j)$, s.t. $\beta_p \geq 0$, $\sum_{p=1}^{K} \beta_p = 1$. This mechanism provides great flexibility since each kernel $\mathbf{k}_p$ can operate on distinct set of features and each feature can be associated with different types of kernels simultaneously. There are several equivalent formulation to MKL, here we adopt [17] for our binary classification. For N data points $(\mathbf{x}_i, y_i)$ where $y_i \in \{\pm 1\}$, $\mathbf{x}_i$ is translated via K mappings $\mathbf{\Phi}_p(\mathbf{x}) \mapsto \mathbb{R}^{D_p}$, $p = 1, \cdots, K$, from the input space into K feature spaces $(\mathbf{\Phi}_1(\mathbf{x}_i), \cdots, \mathbf{\Phi}_K(\mathbf{x}_i))$ where D_p denotes the dimensionality of the p-th feature space. The learning process involves solving the optimization problem,

$$\min \frac{1}{2}\left(\sum_{p=1}^{K} \beta_p \|\mathbf{w}_p\|_2\right)^2 + C\sum_{i=1}^{N} \xi_i, \tag{5}$$
$$w.r.t.\ \ \mathbf{w}_p \in \mathbb{R}^{D_p}, \xi \in \mathbb{R}_+^N, \beta \in \mathbb{R}_+^K, b \in \mathbb{R},$$
$$s.t.\ \ y_i\left(\sum_{p=1}^{K} \beta_p \mathbf{w}_k^T \Phi_p(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\ \sum_{p=1}^{K} \beta_p = 1.$$

This can be converted into a Semi-Infinite Linear Programming (SILP) problem and solved. The optimum weight β can be regarded as a natural indication of the importance of each channel, and hence can be used as the criterion for channel selection. This problem will be further investigated in Sec-5.3.

5. Experiments and Discussion

In this section, we systematically evaluate the effectiveness of our proposed hierarchical spatial-temporal context model on two realistic human action and event databases, *i.e*. the HOHA database of movie videos used in [6] and the LSCOM database of news videos used in [20]. A brief summary of these two databases is provided in Table 1. More details about the databases can be found in [6], [20].

Table 1. A summary of the two databases used for evaluation.

Database	HOHA	LSCOM
# Classes	8	14
# Train Videos	219	3416
# Test Videos	211	about 16000
Source	Movie clips	News video shots
Task	Actions	Actions and Events

These two databases are chosen for evaluation because they exhibit the difficulties in recognizing realistic human actions, in contrast to the controlled settings in other related databases. The problem of recognizing human actions is essentially of multi-label classification, and hence we adopt the one-against-all approach for performance evaluation.

5.1. Results on HOHA Database

In this subsection, we compare our proposed hierarchical spatio-temporal context model with the Space-Time Interest Point (STIP) proposed in [6] for recognizing human actions. To explicitly evaluate the effectiveness of individual model in our proposed framework, we follow the formulation in [6] to employ multi-channel χ^2 kernel for combining multiple channels of features and use greedy search to find out the best combination. The combined kernel function is given as

$$K(H_i, H_j) = \exp\left[-\sum_{c \in \mathcal{C}} \frac{1}{A_c} D_c\left(H_i^c, H_j^c\right)\right], \tag{6}$$

where $H_i^c = \{h_{in}^c\}$ and $H_j^c = \left\{h_{jn}^c\right\}$ are two histograms extracted in channel c for the i-th and j-th samples respectively, whereas $D_c(H_i, H_j)$ is the χ^2 distance, namely

$$D_c\left(H_i^c, H_j^c\right) = \frac{1}{2}\sum_{n=1}^{N^c} \frac{\left(h_{in}^c - h_{jn}^c\right)^2}{h_{in}^c + h_{jn}^c}, \tag{7}$$

and A_c is a normalization parameter set as in [6].

We first individually evaluate the performances of the proposed three levels of spatial-temporal context (not presented here), namely SIFT average descriptor (SIFT), tra-

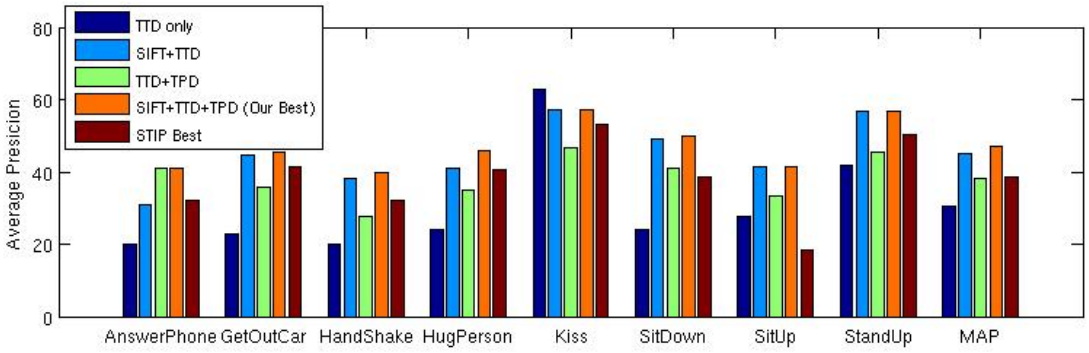

Figure 5. **Detailed performance comparison of our hierarchical spatial-temporal context model with the STIP features in [6] for action classification on the HOHA database**. Each class group from left to right: TTD, SIFT+TTD, TTD+TPD, SIFT+TTD+TPD, STIP.

jectory transition descriptor (TTD), and trajectory proximity descriptor (TPD). Then we evaluate the performances for various combinations: TTD, SIFT+TTD, TTD+TPD, and SIFT+TTD+TPD, and compare the Average Precision (AP) with STIP features in [6]. Note that the latter one reported the state-of-the-art performance on this database.

The detailed performance comparison is shown in Fig. 5, from which we can make the conclusion that our proposed hierarchical spatial-temporal context model (SIFT+TTD+TPD) always yields higher AP performance than STIP features. More specifically, the Mean AP (MAP) is improved from the latest reported 38.39% in [6] to 47.1% based on our proposed new features.

Another observation is that the point-level context (SIFT) and inter-trajectory context (TPD) features can both enhance the discrimination power of the intra-trajectory context features for every class (except for Kiss), and the performance is increased from 30.36% for TTD only to 44.94% (38.17%) when SIFT (TPD) is combined with TTD. The advantage of our proposed features over STIP features can be attributed to the explicit encoding of motion-related information within a relatively long period. The STIP features in [6] also encode the motion information, *i.e*. Histogram of Flow, which is however weak and noisy in characterizing human actions.

5.2. Results on LSCOM Database

On the LSCOM database, the experiments are designed to evaluate the algorithmic capability of our model in recognizing human actions or even more advanced simple events. This database was collected from the TRECV05 Challenge and annotated by humans, and its size is relatively large compared with other databases.

This database was used in [20]. The authors have focused on feature extraction and event description, hence only single-level Earth Mover's Distance (EMD) was adopted. To facilitate a fair comparison, we first use only the channels without grid strategy for experiments on this database. The kernel combination technique listed in Eqn. (6) is adopted to fuse different channels. The 3^{rd} column (denoted as *3CHMUL*) of Table 2 shows these results, from which we can observe that for most classes, our proposed features without grid strategy substantially improve the performance (AP) compared with the algorithm proposed in [20]. The MAP is increased from 25.96% in [20] to 40.67% (by 14.7%). Then we allow all possible grids to our context descriptions, and search for the best combination of channels as explained previously. The results are presented in the 4^{th} column of Table 2, and the performance is further improved over 3CHMUL by 2.4%. Moreover, there are also several classes that our proposed model has not handled well, such as the People-Marching and Exiting-Car. One possible reason is that in such situations, the trajectory information our model exploit is haphazard and hence ruins the performance. This is one point which needs further investigation.

The work in [20] mainly concentrated on using SIFT keypoints to extract motion vectors, but did not go one step further to model the trajectories, which are better representations of motion dynamics and variations. The encoding of transient motion vectors, instead of the sequential relation of motion vector that forms a trajectory, has limited the capability of the algorithm in [20] to solve the action recognition problem.

5.3. Kernel Pruning with MKL

The greedy search scheme for best channel combination as proposed in [6] is attractive because of its non-decreasing nature (the search will terminate if the performance starts to drop). Nevertheless, it could be computationally prohibitive if a number of useful channels are to be selected from a large collection of channels. Formally let the total number of channels be N, and the desirable subset of channels be of size k. Then to seek for a global optimum selection, it costs $C_N^k = \frac{N!}{(N-k)!k!}$ evaluations of single kernel SVMs.

Towards saving the cost of greedy search, MKL has shown good promise. It has been experimentally verified in [17] and several other related works that MKL is effec-

tive in kernel selection. Hence it would be an economic alternative that MKL can roughly select those kernels that are important for discrimination. These roughly selected kernels can then be fed into other module for further processing at a lower cost. More specifically, if we expect to select one best combination from the C_N^k candidates, in our implementation, we only need run MKL on these C_N^k kernels once, and then MKL algorithm automatically selects the few, *e.g.* k_0, possible kernel combinations, and finally we further evaluate the performance of these k_0 for a best candidate. It means we only need run SVMs $1 + k_0$ times, instead of C_N^k times, which greatly saves the time for algorithm evaluation.

Table 2. Performance comparison on LSCOM database. The 14 class labels are 1) exiting-car, 2) hand-shaking, 3) running, 4) demonstration-or-protest, 5) people-crying, 6) walking, 7) singing, 8) riot, 9) dancing, 10) shooting, 11) airplane-flying, 12) election-campaign-greeting, 13) street-battle, and 14) people-marching.

Class ID	Wang *et al*. [20]	Our 3CHMUL	Our BEST
1	**34.8**	12.86	16.8
2	11.4	27.78	**29.76**
3	**68.7**	57.38	61.04
4	36.8	45.21	**47.44**
5	7.2	44.01	**50.19**
6	39.2	46.89	**47.2**
7	11.0	77.14	**79.51**
8	**23.0**	19.45	18.25
9	19.5	61.58	**63.83**
10	12.4	30.37	**35.53**
11	23.1	64.13	**67.22**
12	13.9	38.06	**38.87**
13	32.3	30.95	**32.83**
14	**30.2**	13.60	14.51
MAP	25.96	40.67	**43.07**

6. Conclusions and Future Work

We have proposed a hierarchical framework to encode point-level, intra-trajectory level, and inter-trajectory level spatio-temporal context information of video sequences. This framework has been evaluated on two realistic action and event recognition databases, and has shown superior performance over the state-of-the-art features and algorithms. We are planning to follow three lines in future research. First, to investigate the properties of the three levels of features and their individual merits. Moreover, it is worthwhile to investigate the performance of these features on trajectories generated by other mechanism, *e.g.* KLT. Second, to further extend current model to explicitly account for relative motions, *e.g.* those caused by the frequent camera motions. This kind of motions can create spurious trajectories over frames, which may turn out to be detrimental to action recognition tasks. In the current work we have tried to mitigate the effect of relative motions by limiting the length of trajectories. The third direction is to develop semi-supervised learning algorithm so as to harness the unlabeled data for helping select best channel/kernel or combination.

Acknowledgment

This research is supported in part by IDMPO Grant of R-705-000-018-279 Singapore, and in part as CSIDM Project No. CSIDM-200803 partially funded by the National Research Foundation (NRF) of Singapore.

References

[1] L. Breiman. *Probability*. Society for Industrial Mathematics, 1992.
[2] N. Cuntoor and R. Chellappa. Epitomic Representation of Human Activities. *CVPR*, 2007.
[3] R. Duda, P. Hart, and D. Stork. *Pattern classification*. Wiley New York, 2001.
[4] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. *CVPR*, 2005.
[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. *TPAMI*, 2007.
[6] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008.
[7] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.
[8] Z. Liu and S. Sarkar. Simplest Representation yet for Gait Recognition: Averaged Silhouette. *ICPR*, 2004
[9] D. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.
[10] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004.
[11] E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. *CVPR*, 2005.
[12] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, 2008.
[13] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *IJCV*, 2002.
[14] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. *ICPR*, 2004.
[15] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. *ICRA*, 2001.
[16] Y. Song, L. Goncalves, and P. Perona. Unsupervised Learning of Human Motion. *TPAMI*, 2003.
[17] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *JMLR*, 2006.
[18] P. Tissainayagam and D. Suter. Object tracking in image sequences using point features. *Pattern Recognition*, 2005.
[19] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. *ICCV*, 2003.
[20] F. Wang, Y. Jiang, and C. Ngo. Video event detection using motion relativity and visual relatedness. *ACM Multimedia*, 2008.
[21] Y. Wang, H. Jiang, M. Drew, Z. Li, and G. Mori. Unsupervised Discovery of Action Classes. *CVPR*, 2006.
[22] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. *CVPR*, 2005.
[23] A. Yilmaz and M. Shah. Recognizing Human Actions in Videos Acquired by Uncalibrated Moving Cameras. *ICCV*, 2005.
[24] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV* , 2007.