A Spatio-Temporal Pyramid Matching for Video Retrieval

Jaesik Choi\textsuperscript{a}, Ziyu Wang\textsuperscript{b}, Sang-Chul Lee\textsuperscript{b,**}, Won J. Jeon\textsuperscript{c,**}

\textsuperscript{a}University of Illinois at Urbana-Champaign, 201 N. Goodwin Avenue, Urbana, IL 61801, USA
\textsuperscript{b}Inha University, 1103 High-tech Center, Yonghyun-dong 253, Nam-gu, Incheon, Korea
\textsuperscript{c}Samsung R\&D Center, 3000 Orchard Parkway, San Jose, CA 95134, USA

Abstract

An efficient video retrieval system is essential to find relevant video contents from a large set of video clips, which typically contain several heterogeneous video clips to match with. In this paper, we introduce a content-based video matching system that finds the most relevant video segments from video database for a given query video clip. Finding relevant video clips is not a trivial task, because objects in a video clip can constantly move over time. To perform this task efficiently, we propose a novel video matching called \textit{Spatio-Temporal Pyramid Matching (STPM)}. Considering features of objects in 2D space and time, STPM recursively divides a video clip into a 3D spatio-temporal space like a pyramid and compares the features in different resolutions. In order to improve the retrieval performance, we consider both static and dynamic features of objects. We also provide a sufficient condition in which the matching can get the additional benefit.

\textsuperscript{*}Corresponding author
\textsuperscript{**}Co-corresponding author

Email addresses: jaesik@illinois.edu (Jaesik Choi), wangziyu@inha.edu (Ziyu Wang), sclee@inha.ac.kr (Sang-Chul Lee), won.jeon@samsung.com (Won J. Jeon)

\textit{Preprint submitted to Computer Vision and Image Understanding}  November 11, 2012
from temporal information. The experimental results show that our STPM performs better than the other video matching methods.

**Keywords:** Video retrieval; Query by video clip; High-activity videos; Sport videos; Pyramid matching; Spatio-temporal Pyramid Matching

### 1. Introduction

The convenient access to networked multimedia devices and multimedia hosting services has contributed the huge increase in network traffic and data storage. The recent reports say that 34 percent of the current cell phone users do video recording [1] and video traffic is 40 percent of consumer Internet traffic [2]. In addition to the current video hosting services such as YouTube [3] and Vimeo [4], major IT companies such as Google [5] and Apple [6] have started to offer cloud audio/video storage services to customers.

Considering the recent efforts and deployments of content-based image search such as automatic tagging based on face recognition [7, 8], content-based video search is still under-developed. We have two main observations to explain its shortcomings.

First, The temporal information on videos adds more complexity of dimensions of data, so queries could be more complex than typical text-based ones. In addition, the representations of these queries generated by simple sketch tools [9, 10] are so primitive or generic compared with text-represented queries, they would lead either wrong or diverse query results. More complex querying system (such as dynamical construction of hierarchical structures on targeting videos [11]) requires more elaboration on queries by users, which could be more error-prone.
Second, this has been assumed that the user does not have sample videos at hand for query, so additional querying tools are required. However, this assumption is no longer valid because mobile devices such as digital cameras, PDAs, and cell-phones with camera and solid-state memory enable instant image and video recording which can be used for a video query.

![Diagram of video matching system](image)

Figure 1: Overview of our video matching system: (a) video partitioning for new video entries to database, (b) video matching for a given query video clip

Taking advantage of this opportunity from mobile and ubiquitous multimedia, our content-based video query system takes a sample video clip as a query and searches the collection of videos typically stored in multimedia portal service (such as YouTube, Vimeo, Google Video [12], Yahoo! Video

---

1In this paper, we define the terms as follows - a **scene** is an image frame, a **video clip (or clip)** is a set of image frames in which has continuous movement of objects, and a **video** is a set of clips.
[13], etc.), and suggests similar video clips from the database with relevance evaluation. As shown in Figure 1, our system mainly performs the following two functionalities - (1) offline population of our video database for new video entries to database and (2) online video matching for a new query video. When a video is introduced in the database, it is partitioned into multiple clips by a clip boundary detection based on feature analysis and classification. The partitioned video clips are stored along with metadata information in the database. Next, for a new video from query process, it is analyzed and matched to the stored videos and the relevant scores are calculated by our spatio-temporal pyramid matching system.

The rest of this paper is organized as follows. In section 2, related work on image and video retrieval is discussed. Our spatio-temporal pyramid matching system is presented in section 3. We also analyze the mathematical condition where our spatio-temporal pyramid matching system gets benefits from temporal information in section 4. The experimental results are presented in section 5, and finally we conclude our paper.

2. Related Work

The challenges and characteristics of content-based image and video query systems are well discussed in [14]. A significant amount of research on automatic image annotation [15, 16, 17, 18, 19, 20] has been done, and recently researchers are more focusing on automatic video annotation [21, 22, 23, 24]. Especially, Ulges et al. [25] and Ando et al. [26] discussed video tagging and scene recognition problems, which have similar goals to ours but takes different approaches. The techniques to summarize features in videos using hidden Markov model have been used in [27, 28, 26]. Compared with using hidden Markov model, our modified pyramid matching
kernel has a simpler representation of features in time domain, therefore is faster to calculate the score of relevance feedback for video query.

Recently content-based video retrieval for sports video [29] has been widely discussed. The work in [30] and [31] focused on the framework and personalization of generic sports videos, whereas others target particular sports such as baseball [32, 33], soccer [34], basketball [35], etc.

Different techniques of finding similarity of subsets of video streams have been discussed in [36, 37, 38, 39, 40]. Pyramid matching [41, 42, 43] is known as one of the best matching algorithms for image retrieval and recognition. Our spatio-temporal pyramid matching system [29] extends the spatial pyramid matching [42] to accommodate time domain for efficient video matching and query. Recently, our spatio-temporal pyramid matching is also adapted and improved by follow-up work [44, 45, 46, 47] and shown to be more effective than previous state-of-the-art methods for action retrievals in videos.

A temporally aligned pyramid matching (TAPM) [48] computes the similarity of video clips by dividing a video clip into a set of images, aligning the images independently, and matching the images hierarchically by using the agglomerative clustering [49]. Our STPM is similar to TAPM in a sense that we divide temporal domain hierarchically. However, there is a fundamental difference between STPM and TAPM in a sense that STPM incorporates temporal domain and spatial domains together more tightly. TAPM matches whole images in the lowest level in spatial domain, followed by the temporal hierarchy. Meanwhile, our STPM divides temporal domain with spatial domain. Thus, in the lowest (coarsest) level, we only see local spatio-temporal features, which are shown to be effective in [44]. The size of grid becomes larger along the level of pyramid hierarchy. Moreover, the STPM is much simpler to implement and efficient in computation.
A temporally-binned model [50] and a local spatio-temporal action detector, Space-time interest points (STIP) [51] is also similar to our STPM. The main difference is that our STPM uses a weighted, multiscale (from a local level to the global, video, level) pyramid.

3. System Design

Given a video clip as a new entry to the database, the clip boundary detection in our system divides it into multiple video clips and they are stored in our video matching database. Once the system receives a query video clip, the similarity of the query video clip and the video clips already stored in the database is measured by our Spatio-Temporal Pyramid Matching (STPM) kernel. The measured similarity is used for the rank of video matching. A higher rank of a query and a video clip in the database indicates more similarity.

In this section, we present our definition of STPM kernel and the details of our system, including weight assignment on features, clip boundary detection, and clip similarity matching.

3.1. Spatio-Temporal Pyramid Matching Kernel

Our STPM kernel is an extension of the original Spatial Pyramid Matching (SPM) [42], which is customized to the 2D image matching. Both SPM and STPM stem from the same Pyramid Matching [41]. For STPM, we embrace the time dimension in a sequence of video frames in a video clip to build a 3D pyramid kernel, which is naturally appropriate for video matching. First, an STPM kernel is built by constructing a \((L + 1)\)-level pyramid so that the given video clip sits on level \(L\). Figure 2 shows an example of a pyramid with \(L = 3\). From the top (coarsest) to the bottom (finest)
of the pyramid, the 2-dimensional spatial location and 1-dimensional time dimension are recursively divided into a half. Each divided volume element or *voxel* corresponds to a bag of features and trivially the top-level voxel corresponds to a bag of features of a given video clip.

To compare the similarity of two video clips in the same region\(^1\) \(r\), STPM represents the set of features in each video as a histogram, in which each bin corresponds to a feature, and the height of the bin represents the occurrence of the feature. We refer \(H^l_r(X)\) to the histogram of video \(X\) in region \(r\) at level \(l\). One of the main operations in our video matching is the histogram intersection, which is:

\[
H^l_r(X) \cap H^l_r(Y) = \sum_{f \in F} \min(H^l_r(X)[f], H^l_r(Y)[f]), \tag{1}
\]

\(^1\)A region in this context corresponds to a particular voxel in the pyramid.
where \( \mathcal{F} \) is the set of features (i.e. the set of all bins), and \( H^l_r(\cdot)[f] \) is the height of the histogram \( H^l_r(\cdot) \) for feature \( f \). Figure 3 illustrates three histograms extracted in different levels of the pyramid. Each histogram has counts for features, e.g., diamond and circle in the example. Note that, in level 2 and level 1, a histogram will be calculated for each cubic.

Then, the matching score of video \( X \) and \( Y \) at level \( l \), \( \Gamma^l(X,Y) \) is defined as:

\[
\Gamma^l(X,Y) = \sum_{r \in R^l} H^l_r(X) \cap H^l_r(Y),
\]  

(2)

where \( R^l \) is the set of regions at level \( l \).

The matching is performed progressively from the bottom (level 0) to the top (level \( L \)) of the pyramid. At level \( l \), each dimension is divided into \( \frac{1}{2^l-1} \) regions. In each region, it calculates the histogram intersection, which excludes the features which are already matched in the previous levels. This computation can be done efficiently as follows:
\[ \kappa^L(X, Y) = \Gamma^0(X, Y) + \sum_{l=1}^{L} \frac{\Gamma^l(X, Y) - \Gamma^{l-1}(X, Y)}{2^l} \] (3)

\[ = \sum_{l=0}^{L-1} \frac{1 + 1}{2^l} \Gamma^l(X, Y) + \frac{1}{2^L} \Gamma^L(X, Y), \] (4)

where we call \( \kappa^L(X, Y) \) the \( L \)-level pyramid matching kernel of two video clips \( X \) and \( Y \).

### 3.2. Weight Assignment on Features

![Figure 4: Video matching scheme for Spatio-Temporal Pyramid Matching](image)

For the comparison of video clips which contain constantly moving objects, we choose two features, Shift Invariant Feature Transform (SIFT) [52] and optical flow [53]. SIFT has been widely used to match two still images. However, SIFT features extracted from stationary frames may not good enough to capture the movement of objects in the frames. Thus, we also use the optical flow to match the movement of objects in the sequence of frames in the video clip.
To calculate the similarity of two video clips, we first calculate two STPMs, $\kappa_{SIFT}$ for SIFTs and $\kappa_{OP}$ for optical flows. Then, we use a linear sum of two kernels as follows:

$$\kappa_{STPM}(X,Y) = w_{SIFT}*\kappa_{SIFT}(X,Y) + w_{OP}*\kappa_{OP}(X,Y), \quad (5)$$

where, $w_{SIFT}$ and $w_{OP}$ are tunable parameters and $w_{SIFT} + w_{OP} = 1$. Figure 4 shows $\kappa_{STPM}$ for two video clips $X$ and $Y$.

3.3. Shot Boundary Detection

To detect clip boundaries in videos, we use a boosting algorithm combining weak classifiers which are based on frame-by-frame features. Boosting is a well known method to find appropriate weights for each weak classifier. We follow the established principles, e.g., [54, 55] and use three features, hue channel, entropy of optical flows, and intensity among different possible frame-by-frame features. Intuitively they reflect the nature of scene changes in a video, which means that sharp changes in hue channel, entropy of optical flows, and intensity highly imply the beginning another video clip. Note that other features can be easily pluggable to our framework in different matching scenarios.

3.4. Shot Similarity Matching

When we compare two video clips, typically they have different lengths in time, so selecting the same number of representative video frames from the two videos unlikely happens for video matching. In order to construct a uniform pyramidal structure in temporal domain from a video clip, we extract $m$ frames out of the video clips, where $m$ equals $2^l$. Thus, a video clip is represented as a cube of a $2^l \times 2^l$ spatial region and $2^l$ temporal interval.
Each voxel in the cube includes a set of features, which is represented by a histogram of feature bins.

4. Analysis of Spatio-Temporal Pyramid Matching Kernel

In this section, we analyze a sufficient condition in which temporal information contributes to the video matching.

First, in order for the fair comparison of SPM and STPM for given two video clips, we use the conventional weighted sum of matching scores of key frames for SPM without any temporal information. It is straightforward to prove that the matching score of STPM is equal to or greater than that of SPM. Second, we introduce category gain to represent the gain from temporal information, which is calculated as matching score of STPM minus matching score of SPM for two video clips in the same category (i.e., noisy matching). However, STPM also can introduce noisy gain, which shows the higher matching score between two video clips in different categories. Intuitively, the category gain needs to be greater than the noisy gain to justify the use of STPM. Otherwise, the matching performance of STPM could be worse than that of SPM.

4.1. The Gain of Noisy Matching

The expected noisy matching of SPM and STPM in each level, $\Gamma^l_{SPM}$ and $\Gamma^l_{STPM}$ are calculated from the equation (2). The example is shown in Figure 5. Here we assume that a video frame has $32 \times 32$ grid cells (i.e., 1024 features per video frame), and features are clustered into 200 groups. For frame matching in SPM, a grid cell has one of 200 features. Thus, in the grid cell at the pyramid level $l$, $2^l \times 2^l$ features are aggregated to be matched with the histogram intersection. For the unit cubic voxel at level
expected score of matches per feature increases as the pyramid level becomes higher. Second, the expected score of STPM is higher than the score of SPM in each level. The two observations are caused by the same reason. If the aggregated grid is bigger, the histogram intersection per feature also becomes higher.
For SPM and STPM, the expected noisy matching score is calculated as follows:

\[
\tau^L_{\text{SPM}} = \sum_{l=0}^{L-1} \frac{1}{2^{l+1}} \Gamma^l_{\text{SPM}} + \frac{1}{2^{L-1}} \Gamma^L_{\text{SPM}},
\]

(6)

\[
\tau^L_{\text{STPM}} = \sum_{l=0}^{L-1} \frac{1}{2^{l+1}} \Gamma^l_{\text{STPM}} + \frac{1}{2^{L-1}} \Gamma^L_{\text{STPM}}.
\]

(7)

As shown in Figure 6, the matching score of STPM is higher than that of SPM in every pyramid level. In the minimum pyramid level (0 in the Figure), the pyramid has only one level, so that the matching score is same with the bag-of-features assumption. In the maximum pyramid level (5 in the Figure), the matching score is dominated by the score of \(\Gamma^5\) which is more sensitive to spatial and temporal locations of features.

4.2. Conditions for Superiority of STPM

Suppose \(A\) and \(B\) are two video clips in the same category, and \(A_i\) and \(B_i\) denote \(i\)-th video frame of video clip \(A\) and \(B\), respectively. In addition, \(\kappa^L_{\text{SPM}}(A_i, B_i)\) and \(\kappa^L_{\text{STPM}}(A, B)\), the sound matching scores of SPM (of two video frames \(A_i\) and \(B_i\)) and STPM (of two videos \(A\) and \(B\)). Then, the following equation is normally satisfied:

\[
\kappa^L_{\text{STPM}}(A, B) - \frac{\sum_{i \in I} \kappa^L_{\text{SPM}}(A_i, B_i)}{|I|} > 0.
\]

(8)

Here we refer the value of equation (8) as the gain of STPM compared to SPM. In order to guarantee the better performance of STPM against SPM, the gain should be bigger than the gain of noisy matching, \(\tau^L_{\text{STPM}} - \tau^L_{\text{SPM}}\). Therefore,
\[ \kappa^{L}_{STPM}(A, B) - \frac{\sum_{i \in I} \kappa^{L}_{SPM}(A_i, B_i)}{|I|} > \tau^{L}_{STPM} - \tau^{L}_{SPM}. \]  

(9)

That is, the gain of scores in the same category which is the left hand side of equation (9) needs to be greater than the gain of score in different categories which is the right hand side of equation (9). In other words, \( SPM \) should have room for further improvement by the gain, because the maximum score of \( \kappa^{L}_{STPM}(A, B) \) is 1, which leads the following condition:

\[ 1 - (\tau^{L}_{STPM} - \tau^{L}_{SPM}) > \frac{\sum_{i \in I} \kappa^{L}_{SPM}(A_i, B_i)}{|I|}. \]  

(10)

5. Experimental Evaluation

In this section, we present our experimental settings including the datasets and features that we use, performance criteria for video matching, and experimental evaluations. We use two datasets for benchmarking - (1) USF dataset [56] and (2) sports videos that we collected from YouTube.\(^2\) The performance of video matching with our spatio-temporal pyramid matching is evaluated with two parameters - (1) the quality of video retrieval and (2) the quality of binary decision.

5.1. Datasets

**UCF dataset:** We tested our proposed method with the UCF50 dataset which consists of realistic videos with large variation in camera motion, object appearance, scale, illumination, etc. UCF50 dataset consists of 50 action categories taken from YouTube. Each action category includes 25 groups of video clips. Each group consists of more than 4 video clips.

\(^2\)This dataset is publicly available at http://reason.cs.uiuc.edu/jaesik/viki/supplementary/YoutubeSports/.
Sports videos from YouTube: Several sports highlight videos were selectively collected from YouTube. The contents in the videos are mainly in different sport categories including basketball, football, baseball, and so on. Among these videos, we chose highlights of basketball dunk clips, basketball field goals, and running actions at college football. The highlights are constructed from hundreds of different sports videos, thus our experimental data is not biased towards some specific videos. The videos are divided into several video clips by our clip boundary detection algorithm which is described in Section 3.3. Then, the evaluation dataset which is composed of 200 video clips was prepared by the manual labeling process. In the labeling process, the video clips were categorized based on directions of sporting activities (e.g., running left or right) and camera movements (e.g., focusing up or down). As a result, the number of video clips in each category span from 2 to 36.3

5.2. Features for Shot Similarity Matching

As we described earlier, two representative features, SIFT and optical flows are used.

Dense SIFT: The dense SIFT features are a set of SIFT features which are extracted uniformly in each spatial location. Normally SIFT features are extracted at the salient locations, but dense SIFT features are proven as quite effective in image similarity matching [57, 42]. The SIFT features are clustered into an arbitrary number of groups with k-means clustering. The number of clusters is empirically chosen.

Dense optical flows: In each equally divided grid, an optical flow is

---

3 The details of labels with the dataset are available at http://reason.cs.uiuc.edu/jaesik/viki/supplementary/.
extracted even though the location has no salient feature. For the temporal
dimension, the optical flows are calculated in the original video clips. That
is, the extracted frames in a pyramid level become reference frames. For
each reference frame, its consecutive frame in the original video clip is used
to calculate the optical flows. The flows are also clustered into an arbi-
trary number of groups with $k$-means clustering. The number of clusters is
empirically chosen as well.

In our experiment, we chose 200 and 60 groups for $k$-means for dense
SIFT and dense optical flows and they showed reasonably good performance
than using other values.

5.3. Performance Criteria

**Quality of video retrieval**: The quality of retrieved videos is measured
by the Discounted Cumulative Gain at top $n$ ($DCG@n$) [58] and the Receiver
Operating Characteristic (ROC).

$DCG@n$ is an evaluation metric in information retrieval for measuring
the accuracy of search results, which is given as follows:

$$DCG@n = \sum_{i=2}^{n} \frac{1_{\text{relevant}}(v_i)}{\log_{2}(i)}.$$  \hspace{1cm} (11)

In the equation, $DCG@n$ is the sum of awards for top $n$ retrieved videos,
where $1_{\text{relevant}}(v_i) : \{\text{all videos}\} \rightarrow \{0,1\}$ is an indicator function which
outputs 1 when the retrieved video clip $v_i$ is relevant to the query clip.

We tested our method in receiver operating characteristic (ROC) for the
top $n$ retrieved videos. Given a query video, we retrieved top $n$ relevant
videos and then computed the true positive rate and the false positive rate
as follows:

$$\text{True positive rate} = \frac{|\text{Retrieved relevant videos}|}{|\text{All relevant videos}|}$$
False positive rate \( = \frac{|\text{Retrieved non-relevant videos}|}{|\text{All non-relevant videos}|} \).

**Quality of binary decision:** For evaluation purposes, the video retrieval problem is translated into a binary decision problem. Given a query video clip \( Q \) in a particular category, we choose two video clips, \( A \) in the same category of \( Q \) and \( B \) outside the category. Thus, the similarity score, \( \kappa_{STPM}(Q, A) \) between \( Q \) and \( A \) should be greater than \( \kappa_{STPM}(Q, B) \) between \( Q \) and \( B \). If \( \kappa_{STPM}(Q, A) \) is greater than \( \kappa_{STPM}(Q, B) \), the binary classifier with the matching schema is regarded *correct*. If \( \kappa_{STPM}(Q, A) \) is less than \( \kappa_{STPM}(Q, B) \), the binary classifier is regarded *incorrect*.

Four different schemes are compared in this scenario: (1) \( SPM(SIFT)-I \), a similarity score by matching with SPM on SIFT features extracted from a single key frame of each video;\(^4\) (2) \( SPM(SIFT)-II \), the average similarity score by matching with SPM on SIFT features of several key frames; (3) \( STPM(SIFT) \), the similarity score by matching with STPM on SIFT features; and (4) \( STPM(SIFT+OP) \), the similarity score by matching with STPM on SIFT features and optical flows. Here, we regard \( SPM(SIFT)-I \) and \( SPM(SIFT)-II \) as baseline results. For \( SPM(SIFT)-II \), multiple key frames extracted to match as exactly the STPM did, and then the average of similarity calculated by SPM on the key frames is provided as a matching score of two video clips.

5.4. Experimental Results

**Video query on UCF50 dataset:** We calculate DCG functions and ROC curves for each categories. We average the results out and plot the

\(^4\)We use a single key frame which is \( \lfloor \frac{n}{2} \rfloor \)-th frame out of a \( n \)-frame clip.
Figure 7: DCG@n performance and ROC curves for video clip retrieval in UCF50 dataset.

Results in Figure 7.\textsuperscript{5}

Figure 7a shows that STPM based methods, STPM(SIFT+OP) and STPM(SIFT), outperform the two baselines SPM(SIFT)-I and SPM(SIFT)-II. Figure 7b reports the similar results. The results indicate that matching over temporal features in the STPM helps to improve video retrieval.

Figure 8 shows top 10 categories where the STPM based methods outperform the SPM based ones. As we expected, the STPM based methods are better in the videos with more dynamic actions such as ‘Kayaking’ and ‘Biking’.

Figure 9, on the other hand, shows the bottom 10 categories in which the STPM based methods perform worse than the SPM based ones. Overall, the

\textsuperscript{5}Supplementary results for 100 example queries are available at http://reason.cs.uiuc.edu/jaesik/viki/supplementary/UCF50/.
Figure 8: Top 10 categories where the STPM outperforms the SPM in UCF50 dataset
<table>
<thead>
<tr>
<th>Methods</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>STPM(SIFT+OP)</td>
<td>95.3%</td>
</tr>
<tr>
<td>STPM(SIFT)</td>
<td>94.7%</td>
</tr>
<tr>
<td>SPM(SIFT)-II</td>
<td>91.2%</td>
</tr>
<tr>
<td>SPM(SIFT)-I</td>
<td>77.8%</td>
</tr>
</tbody>
</table>

Table 1: Experimental results of pairwise decisions from randomly selected pairs of samples from UCF50 dataset

STPM based methods maintain the similar performance with the SPM based ones. For example, the SPM(SIFT)-II is better than STPM(SIFT+OP) in only 13 categories out of 50.

**Binary decision on UCF50 dataset**: Table 1 shows the result of binary decisions. The average precision of SPM(SIFT)-I, SIFT matching for a key frame, was 77.8%. With SPM(SIFT)-II, the SPM of multiple key frames, the average precision was improved to 91.2%. Our STPM(SIFT) achieved 94.7% of average precision with only SIFT features. With both SIFT and optical flow features, the average precision was improved to 95.3%. We compared the result on different number of clusters (k=200 and k=60). When $k = 200$, we received a slightly higher previsions ($<2\%$) than when $k = 60$. Thus, we set $k = 200$ in this experiment.

In summary, the STPM achieved higher average precision than the SPM with the same multiple key frames. We can assure that the STPM could find more common features than the SPM, and it is more robust for video classification. Note that, the SIFT features depend on the backgrounds of the videos, so if the background is unique, the SIFT may perform better than the optical flow. We also infer that the STPM performs better when the video has more repeated motions in a fixed scene.

Figure 10 shows two sample video query results upon each query video
Figure 9: Top 10 categories where the SPM outperforms the STPM in UCF50 dataset
Figure 10: An example of video query and results (UCF50 dataset)
Figure 11: DCG@n performance and ROC curves for video clip retrieval in Youtube Sports dataset.

Video query on Sports Videos: Figure 11(a) shows DCG@n and values for three different scenarios for video retrievals in our spatio-temporal pyramid matching - (1) SIFT: using SIFT only, (2) OP: using optical flows only, and (3) SIFT+OP: using combination of weighted SIFT and optical flows.

Figure 12 shows an example query video clip and the matching video clips that the system found. The three video frames showing dunk-clip action on the left column is the sample query video clip. The five sets of three video frames on the right column represent the querying results in ranking order.

Binary decision on Sports Videos: Table 2 shows the result of binary decisions. The precision accuracy of SIFT matching for key frames was 75.7%. With the SPM of multiple key frames, the precision accuracy was improved to 81.9%. The optical flows helped to improve the precision.
Figure 12: An example of video query and results (Sports Video dataset)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPM(SIFT)-I</td>
<td>79.2%</td>
</tr>
<tr>
<td>SPM(SIFT)-II</td>
<td>83.0%</td>
</tr>
<tr>
<td>STPM(SIFT)</td>
<td>84.4%</td>
</tr>
<tr>
<td>STPM(SIFT+OP)</td>
<td><strong>86.3%</strong></td>
</tr>
</tbody>
</table>

Table 2: Experimental results of pairwise decisions from randomly selected pairs of samples (Sports Video dataset)
accuracy up to 84.1%. Our STPM achieved 84.3% of precision accuracy with only SIFT features. With both SIFT and optical flow features, the precision accuracy was improved up to 86.3%.

6. Conclusions

In this paper, we addressed the problem of classifying video clips for content-based video query. The clip boundaries are found using a strong classifier learnt from a boosting algorithm on top of weak classifiers. Then, the similarity of video clips is calculated by our spatio-temporal pyramid matching kernel which includes temporal dimension into the matching schema.

Our experimental evaluation using sports videos and standard UCF50 dataset shows that the temporal dimension is an effective feature to match video clips. We performed a comprehensive comparison with other matching methods and showed that our spatio-temporal matching systems outperformed existing video matching schemes.


in: Proceedings of the 7th European Conference on Computer Vision

[16] J. Jeon, V. Lavrenko, R. Manmatha, Automatic image annotation and
retrieval using cross-media relevance models, in: Proceedings of the
26th Annual International ACM SIGIR Conference on Research and
Development in Information Retrieval (SIGIR’03), 2003.

language model and active learning, in: Proceedings of ACM Interna-

multiple evidence & wordnet, in: Proceedings of ACM International

[19] Y. Gao, J. Fan, X. Xue, R. Jain, Automatic image annotation by incorpor-
ating feature hierarchy and boosting to scape up SVM classifiers, in:
Proceedings of ACM International Conference on Multimedia (MM’06),
2006.

walk with priors on the canonical correlation subspace, in: Proceedings
of ACM International Conference on Multimedia Information Retrieval
(MIR’08), 2008.

[21] S. Feng, R. Manmatha, V. Lavrenko, Multiple Bernoulli relevance mod-
els for image and video annotation, in: Proceedings of IEEE Com-
puter Society Conference on Computer Vision and Pattern Recognition


[28] B. Li, M. Sezan, Event detection and summarization in sports video,


ternational Conference on Computer Vision (ICCV’05), Beijing, China, 2005.


