Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Spatio-Temporal Tube data representation and Kernel design for SVM-based video object retrieval system

Shuji Zhao · Frédéric Precioso · Matthieu Cord

Received: Dec 15th 2009 / Accepted: date

Abstract In this article, we propose a new video object retrieval system. Our approach is based on a Spatio-Temporal data representation, a dedicated kernel design and a statistical learning toolbox for video object recognition and retrieval. Using state-of-the-art video object detection algorithms (for faces or cars, for example) we segment video object tracks from real movies video shots. We then extract, from these tracks, sets of spatio-temporally coherent features that we call Spatio-Temporal Tubes. To compare these complex tube objects, we design a Spatio-Temporal Tube Kernel (STTK) function. Based on this kernel similarity we present both supervised and active learning strategies embedded in Support Vector Machine framework. Additionally, we propose a multi-class classification framework dealing with unbalanced data. Our approach is successfully evaluated on two real movies databases, the french movie "L'esquive" and episodes from "Buffy, the Vampire Slayer" TV series. Our method is also tested on a car database (from real movies) and shows promising results for car identification task.

Keywords Kernel design · Object recognition · Video object retrieval · Spatio-Temporal Tube Kernel

1 Introduction

In the context of video object category classification, tasks become more and more challenging, as some of the 20 "features" from the high-level feature task from TRECVid 2009 campaign illustrates it:

- Classroom: a school - or university-style classroom scene. One or more students must be visible. A teacher and teaching aids (e.g. blackboard) may or may not be visible.
- Person-playing-a-musical-instrument - both player and instrument visible

This work is funded by Rgion le-de-France, project *k-VideoScan* N2007-34HD Digiteo

Shuji Zhao
ETIS Lab, CNRS/ENSEA/Univ Cergy-Pontoise
6, av. du Ponceau, 95000 Cergy-Pontoise, France
E-mail: zhao@ensea.fr

Frédéric Precioso
ETIS Lab, CNRS/ENSEA/Univ Cergy-Pontoise
6, av. du Ponceau, 95000 Cergy-Pontoise, France
E-mail: precioso@ensea.fr

Matthieu Cord
LIP6, CNRS/UPMC
104, av. du Président Kennedy, 75016 Paris, France
E-mail: matthieu.cord@lip6.fr

- Bus: external view of a large motor vehicle on tires used to carry many passengers on streets, usually along a fixed route. NOT vans and SUVs
- Person-riding-a-bicycle - a bicycle has two wheels; while riding, both feet are off the ground and the bicycle wheels are in motion
- ...

The "high level features" or video object categories to be classified and retrieved are not only classes of objects with wide appearance variability (e.g. person, vehicle...), but also classes representing abstract concepts, such as events or actions. In order to handle such complex categorization problems, we need a fast and efficient content-based video retrieval system, which depends on good video object detection, relevant visual features extraction and powerful machine learning techniques. In our work we will consider that the video object detection task is achieved by any recent and very efficient algorithm from state-of-the-art works.

In the framework of retrieving actors in movies, Everingham et al. in [4] proposed to represent an actor by a "face track", represented by a set of face descriptors and clothing descriptors. The matching between two face tracks is based on min distance between the two sets of descriptors and on quasi-likelihoods to obtain posterior probability of matching. In a recent work, Kumar et al. [8] proposed a novel algorithm to find faces in databases of more than 3 millions of images and even distinguish different facial expressions. In [1], Apostoloff and Zisserman extended the descriptors of face track aforementioned with 4 additional facial features and preprocessed the data before the matching process. Furthermore, the matching is not anymore based on evaluating the maximum of posterior probability of a label but on random-fern classifiers. In [6] Guillaumin et al. proposed an approach based on a graph of 13 facial features for single-person retrieval and multi-person naming. In [17] Sivic et al. introduced multiple kernel learning (MKL) based on facial features as in [1,4,6].

In our work, we also consider face tracks in video as the data to represent and classify. However, we propose a framework to get rid off introducing prior knowledge on the structure of video objects of interest. In case of actor face for example, we want to avoid to use facial models to target a more generic representation. We use a SVM classifier combined with kernel similarity functions for the retrieval and recognition task. Instead of considering classical supervised approaches, our kernel-based machine learning method allows us to provide either a supervised classification of the data or, exploiting recent advances in machine learning techniques, an interactive retrieval system, based on active learning strategies.

In our previous work [25], a video object is represented by a set of temporally consistent chains of local descriptors SIFT (a bag of bags of features). A "kernel on bags of bags" is designed to compare the similarity of two video objects. In [24], the video object is represented by a "tubes" of visual features as well as spatial location of features. The design of a new kernel embedding this spatial constraint has been proved to be more powerful for actor retrieval in a real movie. In this paper, we extend the actor retrieval of [25,24] to multi-class object recognition task. Our system is not only applied to category "person", but other category like "car model". We obtain very interesting results for actor multi-class recognition and exhibit the generalization capability of our approach to car model retrieval.

This paper is organized as follow:

- In section 2, we introduce spatio-temporal coherency in the data representation, which considers a video track as an entire object instead of a set of individual images or key-frames [6]. From an object video track, we extract a set of spatio-temporally consistent chains of local descriptors SIFT, that we call a "Spatio-Temporal Tube".
- In section 3, we design a kernel "STTK" dedicated to our "Spatio-Temporal Tube" data representation.
- In section 4, we describe our kernel-based SVM classification framework for both two-class retrieval task and multi-class recognition task. We also describe how to deal with unbalanced training data, which is one of the undesirable problem with SVM machine learning.
- In section 5, our approach is tested on two real movie databases: the movie "L'esquive" and episodes of the TV series "Buffy, the Vampire Slayer", as well as on a car database.

An overview of our system is presented in Fig. 1.

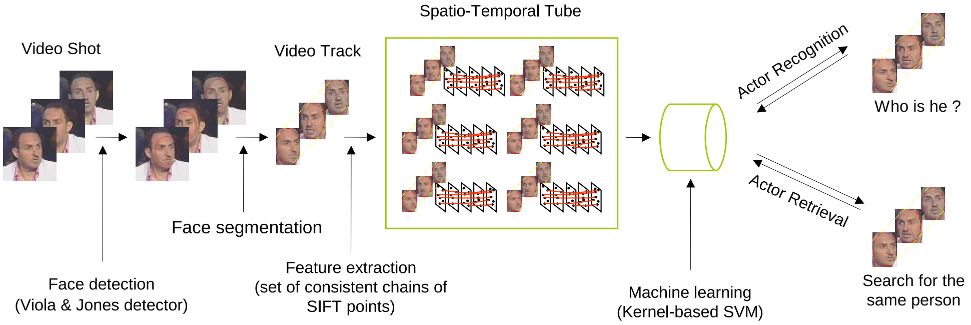


Fig. 1 STTK kernel-based actor learning and retrieval and recognition system.

2 Spatio-Temporal Tube

2.1 Video tracks

For the databases "L'esquive", faces of actors are detected by the algorithm AdaBoost of Viola and Jones [20], extended by Lienhart and Maydt [9], and segmented by ellipses which approximate face contours [25].

For the databases "Buffy", in order to make a fair comparison with the work done by Apostoloff and Zisserman [1], the actor face extraction process for the TV series "Buffy" database is not computed and we directly use face detection and tracking results provided by the authors: face position, scale, frame number, with its ground truth label. In a recent work, Cour et al. [2] showed that a partially labeled data framework could lead to obtain a very large set of such data with ground truth more easily. From the position and scale factor of face region they provided for Episode 2 and 5 of season 5 of TV series "Buffy, the Vampire Slayer", we define an ellipse (instead of the usual rectangle) to approximate the contour of the face. We then extract from each shot containing the face track made of the face regions in the successive frames.

2.2 Scale adaptive SIFT-based features

The first step of the feature extraction process is to detect points of interest and extract SIFT descriptors automatically by Lowe [10] approach in each frame of the face track. One face track is described by a set of vectors (several thousands of vectors for a typical track) where each vector is a 128-dimensional SIFT descriptor representing the 16 8-bin histogram of image gradient orientations inside a 4 x 4 spatial grid centered on one detected SIFT point. Our process of extraction and representation of video object is an unsupervised process, without introducing any model or dedicated facial feature in the region of interest pre-detected.

One of the main issues concerning the extraction of SIFT descriptors with original algorithm [10] lies in the large variation in the size of face images in real movies, which cause large variations of SIFT descriptors in face images, e.g. from several points to several thousand points per image depending on the original image scale. For example, a foreground face track and background face track contain very different information because of the different resolutions. Even in a same face track (of the same actor in the same scene), with "zoom in" and "zoom out" of the camera, two face images in this same track might have a big difference in terms of size (in our experiments it reaches a factor 10). When tracking similar SIFT points along a face track, those variations avoid to match points which should be matched, leading thus e.g.. from several tracked points to several hundred untracked ones. Fig. 2(a) shows example of SIFT feature extraction for three different images representing large-scale, normal-scale and small-scale images. We can see that for large-scale images, we extracted too many small-scale keypoints (that are not reliable features), together with few large-scale keypoints (that are more reliable features). To reduce the irrelevant features and make the algorithm tractable, the scales of images must be reduced. However, if we reduce all the images at a same proportion, there might be no feature extracted from small-scale images, see Fig. 2(b). To solve this scaling problem, we use adaptive scale SIFT extraction, so that we can extract enough SIFT points even if the images are small while reducing the number of irrelevant points extracted in big images, see Fig. 2(c).

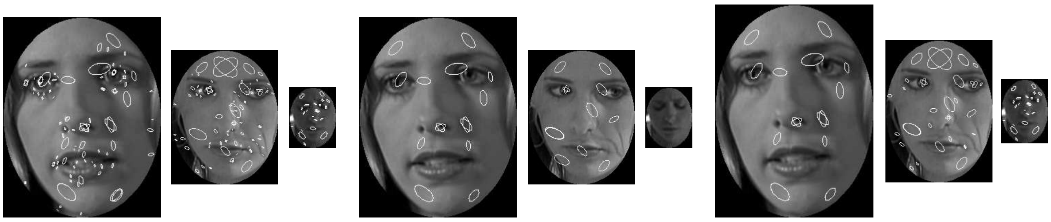

(a) original scale (b) reduced scale (c) adaptive scale

Fig. 2 Comparing fixed scale and adaptive scale for SIFT extraction.

In our work, we use the codes of "SIFT++" of A. Vedaldi [19], which is based on the algorithm of [10], to detect points of interest on face images and extract SIFT feature from these points. The SIFT algorithm is based on a Gaussian pyramid of the input image. The Gaussian image is down-sampled by a factor of 2 after each octave to produce the difference-of-Gaussian images (see [10]). Setting the index of the first octave "first-octave" to $n = -1, 0, 1, 2...$ make the base of the pyramid to be 2^{-n} times of the input image, e.g. -1 corresponding to two times larger than the input image. We make the "first octave" parameter adaptive to the scale of the face image by selecting n that limits the scale of the first octave within a certain range (50 to 100 pixels in width).

We extract also the spatial position of each SIFT points in the track, then normalize it with respect to the size of the face image, to finally enrich SIFT points with their relative position in the track.

2.3 Optimized Spatio-Temporal feature extraction

An ideal face track should contain only consistent information to process face recognition. However, since we do not preprocess the face tracks, the relevant SIFT points, present in almost each frame of the track (for instance, SIFT on the nose, on the eyes or on a scar...), are mixed up with many other SIFT points which are artefacts as lighting changes, occlusions (hair, glasses...), or video compression blocks, etc. In order to clean up these false points of interest, a tracking process assuming the spatio-temporal coherency of relevant visual features in the face track is used to eliminate non-persistent points in the face track.

One of the classic approach of SIFT points matching is to find the 2 nearest neighbors of each keypoint from the first image among those in the second image, and only accepting a match if the distance to the closest neighbor is less than 0.6 of that to the second closest neighbor (see [10]). In this paper, we propose successive frame spatio-temporal coherency features tracking strategy for SIFT points matching. The tracking is done by selecting from two consecutive frames the 40 pairs of best matched points with feature similarity (L^2 distance of SIFT vectors below 150) and spatial proximity (relative position below 0.2) and link them into chains. See red lines of Fig. 5. There are two advantages of our tracking approach: First, we consider the spatio-temporal coherency of relevant visual features of successive frames in the face track, because matching for successive frames is more stable than matching for isolated images, as illustrated in Fig. 3; Second, as two successive frames in a track are very similar, our matching of keypoint compute only the distances between keypoint of similar position and similar scale (relative to face scale), that makes our algorithm much more efficient than matching by computing all the couples of keypoints of the two images.

The tracking is based on the L^2 distance between pairs of SIFT descriptors from two consecutive frames; we will call this distance "SIFT distance". In order to match the relevant points in a track, we should compute for each couple of consecutive frames, the SIFT distances between all the possible pairs of SIFT descriptors and fill a matrix as shown in Fig. 4(a). In order to reduce the computation time, we assume that two SIFT

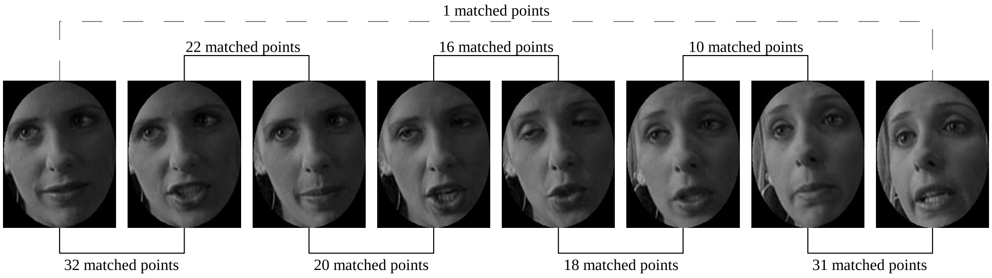


Fig. 3 Comparing of SIFT matching for isolated images and for successive frames. The numbers of these eight frames are $10156, 10159, 10163, 10174, 10176, 10178, 10181$ and 10183 respectively.

descriptions of a temporally persistent point must be very similar in scale and position. Thus, we sort the SIFT vectors by ascendant order of scale, then we do not compute the entire matrix as presented in Fig. 4(a) but we consider only pairs of SIFT points, of similar scale, whose SIFT distance will be around the diagonal (10% of the size of the matrix). We effectively compute the SIFT distance for remaining pairs of points with a position difference lower than 20% of the size of the face images, Fig. 4(b). From the remaining pairs of SIFT points in the sparse matrix, we first keep pairs of points whose SIFT distance is lower than an experimentally determined threshold (fixed to 150, which are selected by cross-validation on the training set tracks of “L’esquive” database). Then, among those remaining pairs, we consider up to 40 of the best matching pairs. The resulting matrix is presented in Fig. 4(c) where the remaining points represent the tracked SIFT points between two consecutive frames.

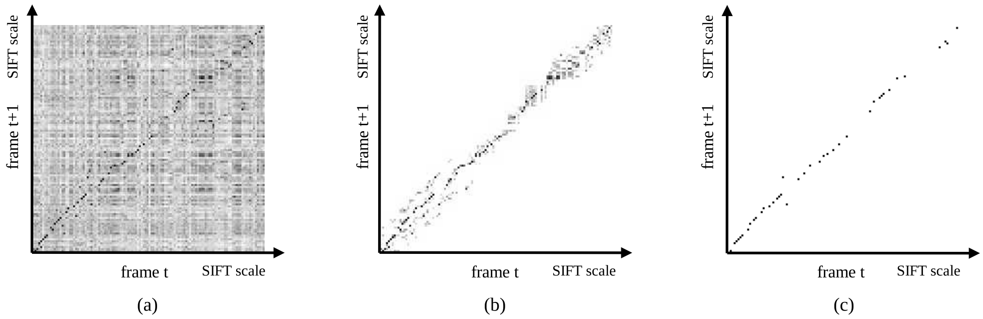


Fig. 4 Matrix of distances for two consecutive frames (black points: nearest SIFT descriptors) (a) full matrix; (b) sparse matrix; (c) selected matrix

All the pairs of matched SIFT points are linked into chains. These chains of matched SIFT points, obtained from a face track, that we call a “tube”, hence represent the temporal coherent support of face information. Such approach for SIFT point tracking is quite economic, from a time processing point of view.

Nevertheless, this first alignment of SIFT points is not relevant enough because some points of interest are disappearing in some frames. In fact, when tracking similar SIFT points along a track, the condition variations (scale, luminance, position, etc.) avoid to match points which should be matched, leading thus an abundance of short chains in a tube. This causes redundant data representation and expensive computing.

We then propose “intra-tube chains tracking” technique [24] to obtain more consistent and more compact chains extracted from each video track while reducing the number of chains and thus reducing computational complexity. See green line of Fig. 5. The intra-tube tracking is achieved by matching two short chains through their average SIFT vectors (L^2 distance below 200 in our case) and the relative average positions (below 0.3 in our work). The matching chains are then considered as the same point tracked on the face and linked into a long chain. Thus, the chains of SIFT descriptors become more consistent and the number of chains per tube

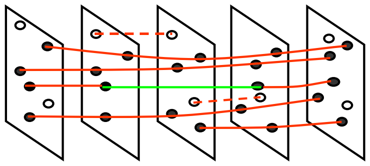


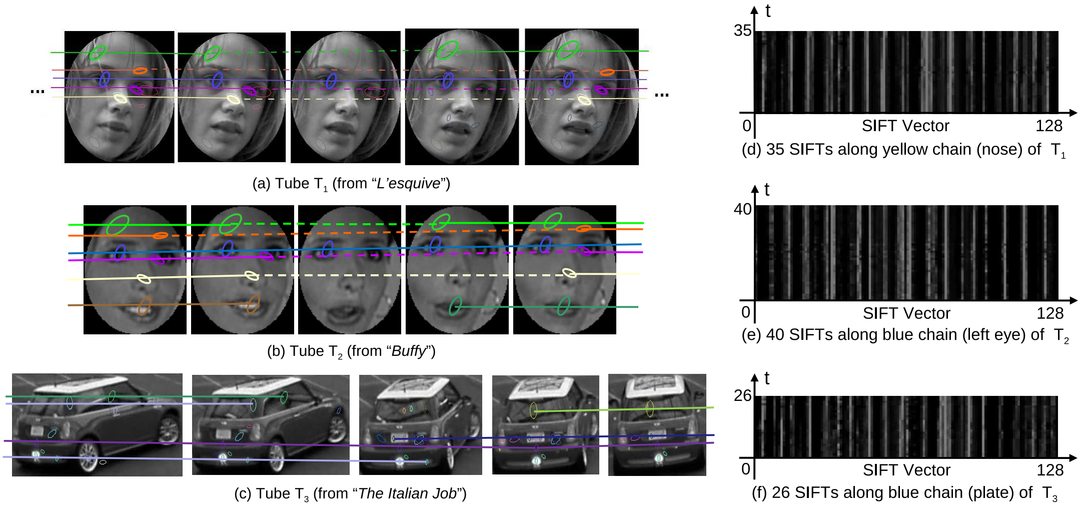
Fig. 5 Intra-tube chain tracking. (Solid lines: consistent chains, dash lines: noise, green lines: link of two short chains)

is highly reduced. Such process provides quite stable SIFT points in the feature space as shown on Fig. 6(b) (one line represents a 128-dimensional SIFT vector while in column you can see the variation of one of these 128 values along the chain).

2.4 Spatio-Temporal Tube data representation

As a result of our spatio-temporal coherent feature tracking, a video object is represented by a tube of consistent chains of SIFT descriptors. To better represent this structural visual information, we introduce the position of each SIFT point in the representation of points in the tube, so that comparisons of chains in "same areas" (of a face) have a stronger impact on overall similarity than comparisons of chains from different areas.

We concatenate spatial positions after 128-dimension description of each SIFT to obtain 130-dimension vectors and to provide tubes containing rich visual information, that we call "Spatio-Temporal Tube". Three examples of Spatio-Temporal Tubes are shown in Fig. 6 (a), (b) and (c), while Fig. 6 (d), (e) and (f) illustrate the temporal stability of SIFT descriptors along its tracked chain.



(a) Tube T_1 (from "*L'esquive*")

(d) 35 SIFTs along yellow chain (nose) of T_1

(b) Tube T_2 (from "*Buffy*")

(e) 40 SIFTs along blue chain (left eye) of T_2

(c) Tube T_3 (from "*The Italian Job*")

(f) 26 SIFTs along blue chain (plate) of T_3

Fig. 6 Spatio-Temporal Tube, SIFT points along the same chain are of the same color (a)(b)(c) examples of two face tubes and a car tube; (d)(e)(f) stability of SIFT points along three chains.

3 Kernel design for Spatio-Temporal Tube

In our work, we design a kernel dedicated to our data representation, in order to compare the similarity of two face video tracks, which are represented by two spatio-temporal tubes of features. This kernel is called: Spatio-Temporal Tube Kernel (STTK).

Let us denote T_i a tube, C_{ri} a chain of V_{lri} "SIFT" vectors and V_{lri} a 130-dimension vector (128-dimension SIFT vector S_{lri} + 2-dimension spatial position P_{lri}). Using set formulation: $T_i = \{C_{1i}, \ldots, C_{ki}\}$ and $C_{ri} = \{V_{1ri}, \ldots, V_{pri}\}$, We want to design a kernel function $K(T_i, T_j)$ which will represent the similarity between two tubes.

As presented in section 2, SIFT vectors from the same chain are spatio-temporally consistent. To reduce the amount of data to be processed, we propose to factorize the SIFT tracked chains by representing each chain C_{ri} with a unique vector $\overline{C_{ri}}$: the mean of all the SIFT descriptors along this chain. It has to be noticed here that the factorization process of SIFT tracked chains is not a simple averaging process of all SIFT vectors along an interesting point into one mean SIFT vector. Indeed, although SIFT points are tracked along object track in order to provide more temporally consistent chains, "the same" interesting point along object track can lead to several chains depending on the variability of its SIFT description in this track or view angles tolerated for the object, etc. This is illustrated on Fig. 6(b) with, for instance, a SIFT point extracted on Buffy mouth: brown chain "becoming" dark green chain. We want to separate SIFT description from spatial position in the "SIFT" vector V_{lri} in order to better handle each one of these features. This factorization and separation process is achieved through two mapping functions:

$$\phi_f(C_{ri}) = \frac{1}{p}\sum_{l=1}^{p} S_{lri} = \overline{S_{ri}} \tag{1}$$

providing the mean SIFT vector and

$$\phi_p(C_{ri}) = \frac{1}{p}\sum_{l=1}^{p} P_{lri} = \overline{P_{ri}} \tag{2}$$

providing the mean position vector.

We can prove that the similarity functions we are designing are valid kernels, using definitions, proofs and properties on kernels from Chapter 3 in [16].

Proved it exists an embedding function $\Phi : \mathbb{T} \to \mathbb{H}$, which maps any tube T_i to $\Phi(T_i)$ in a Hilbert space $\mathbb{H}$, one can define the kernel on bags K by a dot product in the induced space $\mathbb{H}$:

$$K(T_i, T_j) =< \Phi(T_i), \Phi(T_j) > \tag{3}$$

The "power" similarity function between bags K, is a kernel function if the similarity function between chains k_c is a kernel:

$$K(T_i, T_j) = \sum_r \sum_s \frac{|C_{ri}|}{|T_i|} \frac{|C_{sj}|}{|T_j|} {k_c}^q(C_{ri}, C_{sj}), \tag{4}$$

where $|C_{ri}|$ represents the length (number of frames) of the chain C_{ri}, $|T_i|$ represents the length of the tube T_i. Indeed, if ${k_c}^q$ is a kernel, by definition there exists a mapping function ϕ_c such that:

$${k_c}^q(\mathbf{x}, \mathbf{y}) =< \phi_c(\mathbf{x}), \phi_c(\mathbf{x}) > .$$

Thus, we can rewrite K as:

$$K(T_i, T_j) = \sum_r \sum_s \frac{|C_{ri}|}{|T_i|} \frac{|C_{sj}|}{|T_j|} < \phi_c(C_{ri}), \phi_c(C_{sj}) > .$$

Then, using dot product bilinear properties:

$$K(T_i, T_j) =< \sum_r \frac{|C_{ri}|}{|T_i|} \phi_c(C_{ri}), \sum_s \frac{|C_{sj}|}{|T_j|} \phi_c(C_{sj}) >,$$

which is still a dot product. We can then define a new mapping function such that:

$$\Phi(T_i) = \sum_r \frac{|C_{ri}|}{|T_i|} \phi_c(C_{ri}).$$

Thus, if ${k_c}^q$ is a kernel then $K(T_i,T_j)$ is a kernel.

Let us focus now on the similarity function on “SIFT” Chains:

$$k_c(C_{ri},C_{sj}) = k_f(\phi_f(C_{ri}),\phi_f(C_{sj}))k_p(\phi_p(C_{ri}),\phi_p(C_{sj})), \tag{5}$$

where k_f is the feature similarity function between mean SIFT vectors of each chain and k_p is the position similarity function between mean position vectors of the same two chains. If k_f and k_p are kernels over $\mathscr{X}$x$\mathscr{X}$ then the product $k_f k_p$ is also a kernel. Thus, if k_f and k_p are kernels, k_c is a minor kernel and ${k_c}^q$ in Eq.4 also.

Using the power q in the definition of K leads to a good approximation of max function which, despite the claim in [21], turns out to be false since max function is actually not positive definite as demonstrated by Siwei Lyu in the appendices of [11]. For this reason, from a theoretical point of view, it is not safe to use max function as a kernel in an SVM. However, from a practical point of view, it might still achieve good performances.

We have then to show that k_f and k_p are two kernel functions over (C_{ri},C_{sj}): We use the Gaussian χ^2 kernel for the feature similarity function on chains:

$$k_f(\phi_f(C_{ri}),\phi_f(C_{sj})) = \exp\left(-\frac{1}{2\sigma_1^2}\chi^2\left(\phi_f(C_{ri}),\phi_f(C_{sj})\right)\right). \tag{6}$$

Let us remind that if ϕ is a mapping over $\mathscr{X}$ and if k is a kernel over $\mathscr{X}$x$\mathscr{X}$ then $k(\phi(\mathbf{x}),\phi(\mathbf{y}))$ is a kernel function. Thus, using ϕ_f the mapping function defined in Eq.1, k_f in Eq.6 is a kernel function.

The position part of the minor kernel on chains k_c is defined as the following similarity function on the relative positions of the chains:

$$k_p(\phi_p(C_{ri}),\phi_p(C_{sj})) = \exp\left(-\frac{\|\phi_p(C_{ri})-\phi_p(C_{sj})\|^2}{2\sigma_2^2}\right) = \exp\left(-\frac{(\mathbf{x}_{ri}-\mathbf{x}_{sj})^2+(\mathbf{y}_{ri}-\mathbf{y}_{sj})^2}{2\sigma_2^2}\right), \tag{7}$$

where $(\mathbf{x}_{ri},\mathbf{y}_{ri}) = \phi_p(C_{ri})$ is the mean position of SIFT points along chain C_{ri} of tube T_i. Combined with the mapping function ϕ_p defined in Eq.2, k_p in Eq.7 is a kernel function.

Let us remind that the position of a SIFT point is normalized by the size of the face image. The position part k_p introduces the importance of the comparison between two chains approximately at the same position, e.g. left eye chain of tube T_i and left eye chain of tube T_j. For the comparison between two chains at much different positions, e.g. left eye chain of tube T_i and mouth chain of tube T_j, the weight is reduced. Thus, the importance of this matching in the evaluation of the similarity is also lowered.

One second effect of the kernel of Eq.4 that can cause problems is the influence of the the “size” of the tube (the length of the video track and the size of face images in the track). Since the kernel is defined by form of “sum”, the “bigger” the tube, the more chains it contains, hence the higher the value of the kernel concerning the tube. For example, the kernel value between a tube of 100 chains and any other tube is almost always higher than that between a tube of only 5 chains and any other tube. To remove this effect, one technique is to normalize the kernel as described in Chapter 3 of [16]:

$$K'(T_i,T_j) = \frac{K(T_i,T_j)}{\sqrt{K(T_i,T_i)\cdot K(T_j,T_j)}}. \tag{8}$$

After normalization of kernel, the similarity of each tube with itself is always 1, however large its “size”. Again, the proof that K' in Eq.8 is a kernel can be found in Chapter 3 of [16].

4 Multi-class STTK SVM for object retrieval and recognition

We use a kernel-based SVM as classifier: the SVM is a robust and powerful classification technique for two-class problems when data can be linearly separated; kernel functions enable the linear separation of data, which cannot be linearly separated in the original feature space, by projecting them into a Hilbert space of higher dimension.

For the multi-class classification task, we use the one-vs-all strategy, where N binary SVMs are trained for solving a N-class problem. For each one of the q classes, we train a binary SVM classifier and get the decision function:

$$u_q(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{9}$$

The training process (finding the optimal α_i) corresponds to a QP problem that we solve with the classic SMO algorithm [14] which terminates when all of the Karush-Kuhn-Tucker (KKT) optimality conditions are fulfilled. These conditions introduce a global bound C on α_i values which set the trade-off between the largest possible margin between examples and the number of errors allowed.

Then we normalize $u_q(\mathbf{x})$ into $R_q(\mathbf{x}) \in [-1, 1]$, and defined it as the "relevance" of each example. Labels are assigned to data according to the highest relevance among the N SVMs:

$$n = \arg \max_{q=1,\dots,N} \{R_q(\mathbf{x})\}. \tag{10}$$

If $R_n(\mathbf{x})$ is over the confidence threshold, the label n is assigned to $\mathbf{x}$.

One of the problems of the multi-class SVM classification is often to deal with unbalanced datasets where negative examples far outnumber positive examples. An unbalance training dataset could cause the excursion of separation boundary.

Many approaches have been used to deal with the unbalanced data. Biased Penalties [12,13,22] consider different error costs for the positive (C^+) and negative (C^-) classes instead of the global classic cost C mentioned previously. In our experimentation we use the Biased Penalties proposed by Morik et al. [12]:

$$\frac{C^+}{C^-} = \frac{N^-}{N^+}, \tag{11}$$

with N^+ the number of positive samples and N^- the number of negative samples.

5 Experiments

We have tested our STTK-based object retrieval and recognition system on three real movie databases and obtained interesting results: (1) TV series "Buffy, the Vampire Slayer" with same tracked ground truth data as in [1] to compare the performance of our actor recognition framework with facial feature based approach and key-frame based approach, to test our method of balancing unequal classes, to evaluate our adaptive SIFT feature extraction and to evaluate our system against occlusion; (2) Movie "L'esquive" to compare the performance of our actor retrieval framework with our previous works [24,25]. (3) "Car model" database to evaluate the capacity of our system to be applied to other video object categories.

From the face tracks (respectively car tracks) of these three databases, we extract sets of spatio-temporally coherent features, tubes of consistent chains of SIFT descriptors, with the mean normalized position of SIFT points in each chain. These visual features have been used as input to the retrieval system RETIN [5], with our STTK kernel for SVM core.

5.1 Actor recognition on TV series "Buffy"

The database "Buffy" consists of episodes 2 and 5 from season 5 of the TV series "Buffy, the Vampire Slayer", and contains 2462 tracks over 12 actors. The tracks vary in length from 1 to 404 frames, and there are 53032 labeled face detections in the database.

The two experimental scenarii on "Buffy" are precisely the ones defined in Apostoloff and Zisserman work [1]: the first is the intra-episode recognition, we train our classifiers on the 159 training tracks of episode 2 season 5, and test them on all other tracks of the same episode that are at least 10 frames long (constraints from [1]); the second is the inter-episode recognition, we used the 533 tracks from episode 2 season 5 (training tracks and testing tracks of the first scenario) to train the classifiers and then test them on the other episode, episode 5 season 5, which contains 482 tracks of at least 10 frames.

The average number of SIFT chains extracted from a track is 68, varying from 1 to 253. We then put these tubes of SIFT vectors into our multi-class SVM machine learning system for object recognition using our STTK kernel functions of Eq.4, 5 and 8. In our work, we set parameters $q = 2, \sigma_1 = 3, \sigma_2 = \sqrt{0.05}$, which are selected by cross-validation on the training set tracks from episode 2 season 5 of "Buffy" database.

5.1.1 Evaluation of STTK-based approach

We use the test set tracks from episode 2 season 5 of "Buffy" movie to: evaluate adaptive SIFT feature extraction, test our method of balancing unequal classes, and compare our system with key-frame based approach.

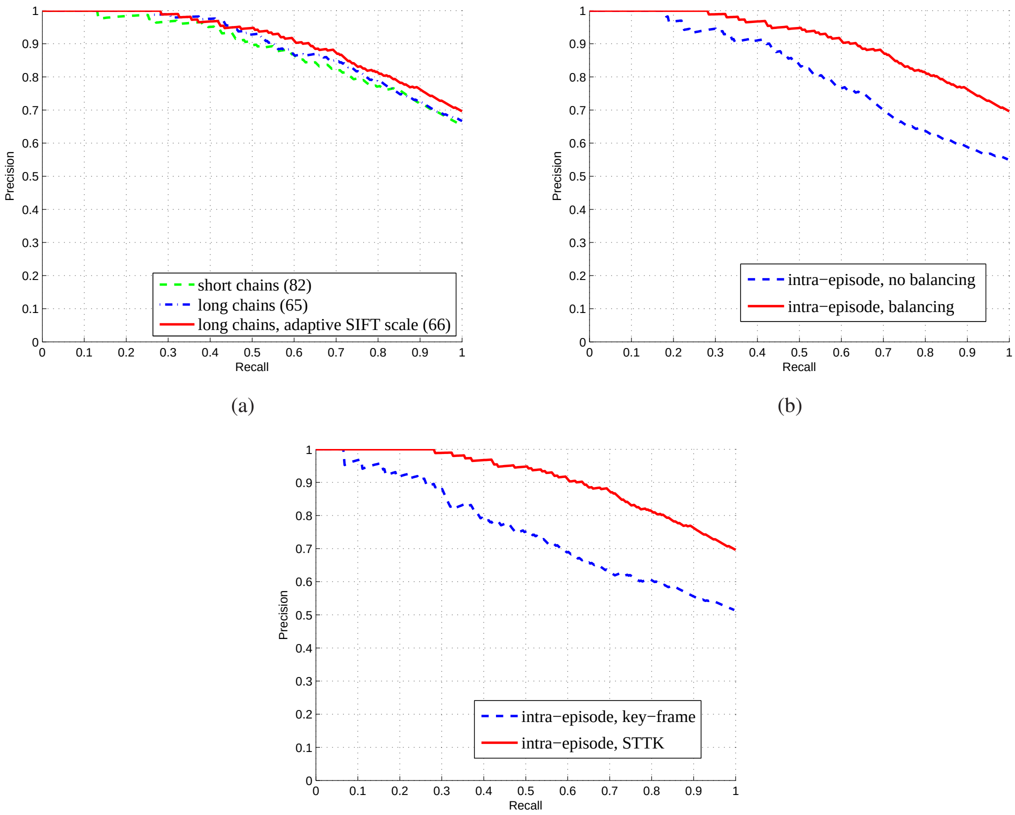
For the evaluation of our adaptive SIFT feature extraction, we test on different way of extraction: with or without intra-tube tracking as described in section 2.3, with fixed scale (first-octave $n = 0$) or with adaptive scale (first-octave $n = ... - 1, 0, 1, 2...$) as described in section 2.2. As showed in Fig. 7(a), the intra-tube tracking process obtains more consistent chains extracted from each video track, thus increases the precision while reducing the amount of data (number of chains from 82 to 65). With scale-adaptive SIFT feature extraction, we extract almost the same average number of chains per tube (from 65 to 66) while improve the precision because of the enhancement resolution for minor images and the reduction of noise for larger images.

For the evaluation of balancing unequal classes, we test respectively with same penalties C (no balancing) and with Biased Penalties C (balancing). Fig. 7(b) shows the difference of before/after balancing training data with Biased Penalties. The precision/recall curve of balanced training set is much better than that without balancing process.

To compare with key-frame based approach for recognition, we select manually the best representative image for each track of episode 2 season 5 of "Buffy". We test on the database and select best parameters for key-frame based approach. We use same parameters of SIFT extraction as STTK based approach to extract SIFT features from the key-frame of each track. The average number of SIFTs in a tube is 48, varying from 9 to 117. We use the same configuration of kernel of that in STTK-based approach replacing two chains C_{ri} and C_{sj} by two SIFT vectors S_{ri} and S_{sj}. See Fig. 7(c) for the comparison of key-frame based and STTK based SVMs. The STTK based system is much better than the key-frame based one, illustrating that the factorization of SIFT chains is not equivalent to a key-frame based approach.

5.1.2 Comparison with facial features based approach

For the comparison with the Random-ferns approach proposed by Apostoloff and Zisserman [1], which is one of the facial features based approaches, we test our multi-classes actor recognition system on two same scenarii as in [1]: intra-episode and inter-episode actor recognition. Our precision/recall curves of intra-episode and inter-episode are showed in Fig. 8 (continuous red lines). We have extracted Precision of Random-ferns approach for several Recall rates from the curves of [1] and reported in Table 1 which illustrates that our approach performs better than Random-ferns approach. As explained in [1], the intra-episode performs better than inter-episode due to the "ABAB shots" that are present in the same episode, e.g. two alternate view angles during a face to face conversation.



(a)

(b)

(c)

Fig. 7 Precision/recall curves for actor recognition on "Buffy" database. Recall is the proportion of tracks assigned labels at a given confidence level, and precision the proportion of correctly labelled tracks.(a) evaluation of adaptive SIFT extractions and intra-tube tracking; (b) evaluation of balancing; (c) comparing key-frame based and STTK based approach.

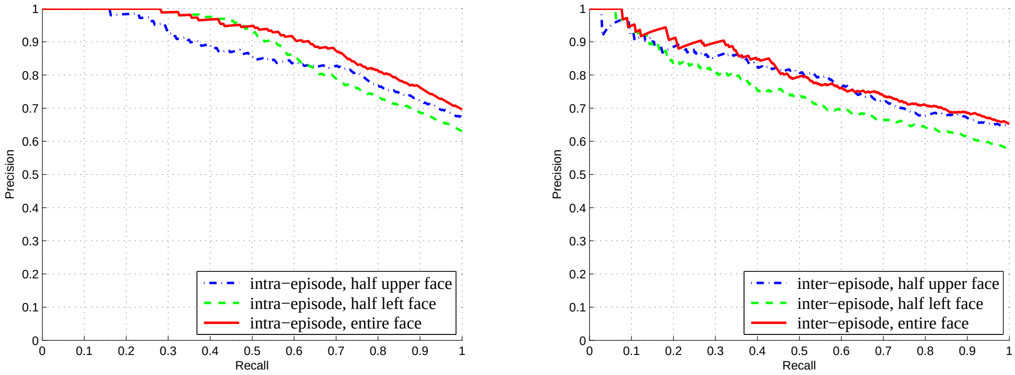
Table 1 Quantitative precision results at different levels of recall.

	Intra-episode					Inter-episode				
Recall	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
Random-ferns	0.98	0.94	0.78	0.68	0.6	0.9	0.8	0.75	0.65	0.55
Proposed method	1	0.97	0.91	0.81	0.7	0.91	0.85	0.76	0.71	0.65

A sample set of identification results of our STTK based actor recognition system is showed in Fig. 9.

5.1.3 Robustness of data representation to occlusion

Facial occlusion is one of the challenging problem for face recognition. It has been already shown in [3] that, given a registered face image, the facial occlusion causes only a small drop in local approaches. To illustrate the robustness of our approach against occlusion, we create image with different occlusion: "half upper face" images and "half left face" images, see Fig. 10. We use entire faces to train our system then compare multi-class precision/recall of recognizing entire faces, respectively with upper half faces and left half faces. The curves of Fig. 8 show that our system can overcome the problem of occlusion with a slight loss of precision. This robustness to occlusion or partial data extraction is one explanation of our choice not to consider facial



(a) (b)

Fig. 8 Precision/recall curves for actor recognition on “Buffy” database: (a) intra-episode, (b) inter-episode.

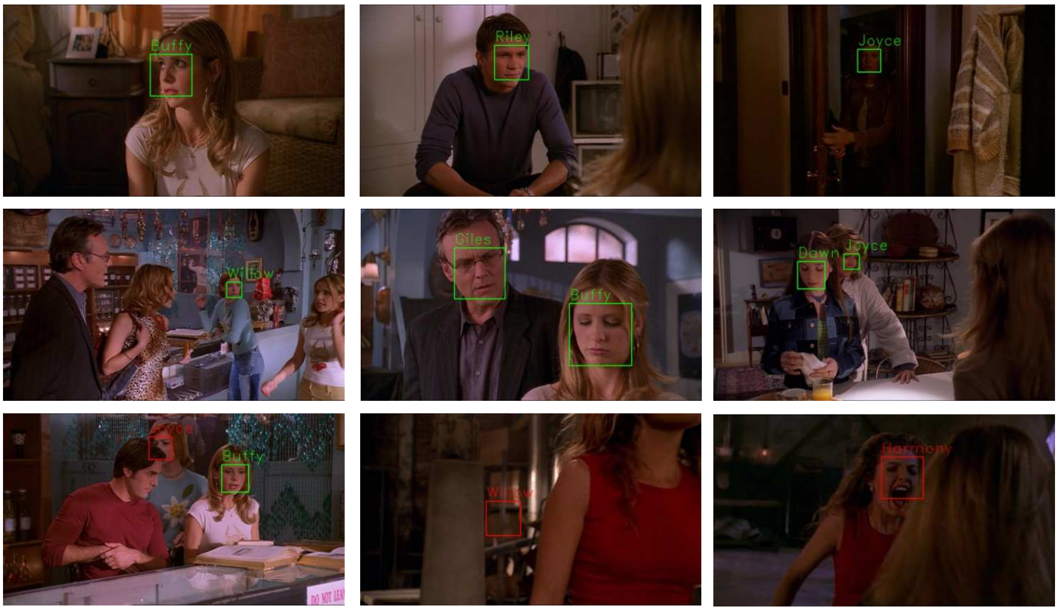


Fig. 9 Sample identifications from episode 05-05. Green squares mean correctly matched faces, while red squares mean failure cases.

features at first. Furthermore, this data representation is also more generic as the next experiments illustrate it.

5.2 Active learning for object retrieval

This experiment exhibits the two main interesting properties of our framework. First, designing a kernel function allows us not only to consider a recognition context as presented before but also to take benefit from recent advances in machine learning techniques to propose an interactive video object retrieval system. Second, our data representation and our STTK-based SVM retrieval system shows high potential of generalization to other kind of video objects. Active learning is a powerful machine learning technique to incrementally build the training set instead of preprocessing it. The high generalization capability and good

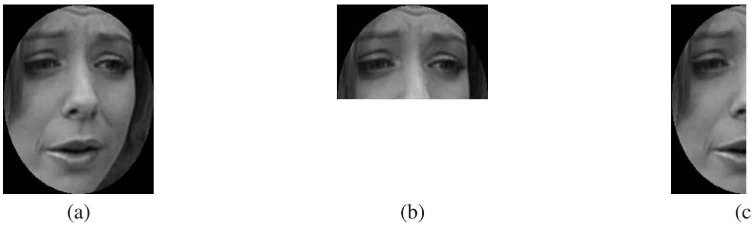


Fig. 10 Test on occlusion: (a) entire face; (b) half upper face; (c) half left face.

user adaptability of active learning made this technique attractive as well for binary object recognition tasks [7], as for relevance feedback in video [23]. Several strategies have been proposed to iteratively either minimize the error of generalization [15] or focus on most uncertain data [18] in order to increase the size of the training set.

5.2.1 Comparison with previous work on Film "L'esquive"

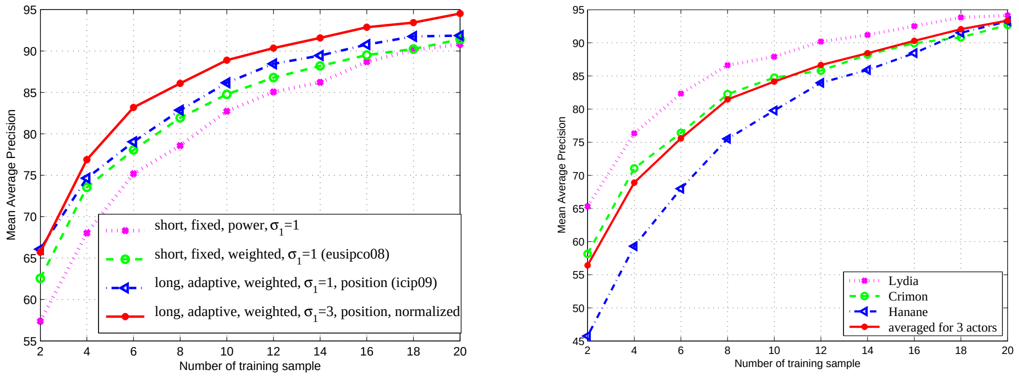
In order to detect faces from the french movie "L'esquive", we use the extension of Lienhart and Maydt [9] of the popular face detection algorithm based on AdaBoost proposed by Viola and Jones [20], implemented in the OpenCV library. Faces are segmented by ellipses which approximates face contours [25]. This face detection algorithm is actually the same as the one used in [1] and thus previous experiments on Buffy database except that we do not perform any face tracking after detection. The database "L'esquive" contains 200 face tracks of 11 actors, with 54 images of face in each track on average. From the 200 tacks of the database "L'esquive", we extract tubes of chains of SIFT descriptors. The number of chains extracted in a tube is either 169 (varying from 19 to 741) with short chains and fixed scale for the "first-octave" as in [25], or 64 (varying from 16 to 260) with long chains and adaptive scale for the "first-octave" as in [24]. We then put these tubes of SIFT vectors into our machine learning system using our STTK kernel functions of Eq.4, 5 and 8. In our work, we set the same parameters as for actor recognition ($q = 2, \sigma_1 = 3, \sigma_2 = \sqrt{0.05}$).

For the comparison with previous works, we train a binary SVM classifier for each actor on "L'esquive" database with few examples: 2, 4 ... 20 examples are picked up randomly in the whole database but preserving balance between positive and negative examples. The Mean Average Precision (MAP) on the whole database of 200 tracks is computed for character "Lydia" (as in previous works) to evaluate the improvements of introducing adaptive scale SIFT feature extraction, optimizing parameter σ_1 and normalizing the kernel.

From Fig. 11(a) we can see that with the adaptive scale SIFT feature extraction, we have more information for minor tubes and less noise for larger tubes, and thus obtain improvements on Mean Average Precision, and reduce the average number of chains per tube from 169 to 52. The kernel normalization (which can deal with the great variation of chains numbers in the tubes) and the optimization of parameter σ_1 (which was set to 1 in previous works by cross-validation on training set from "L'esquive" database) by cross-validation on the larger training set from episode 2 season 5 of "Buffy database", improves about 3% on Mean Average Precision. On Fig. 11(a) we show that our STTK system incrementally learns the 3 most significant classes from "L'esquive database".

5.2.2 Car model active retrieval

We test our STTK-based object retrieval system on a database of car tracks containing 3 car models (volkswagen beatle, volkswagen new beatle and mini cooper S), extracted by hand, from different movies "The Italian Job" (both versions 1969 and 2002 remake) and "Herbie: Fully Loaded" (2004). We make the assumption, as in many previous works [1,4,6], that an object detection algorithm is provided. The object detection step is thus seen as a data pre-processing, and we focus on the objet retrieval task. In order to extract enough car tracks, the movies we chose are movies in which cars get some kind of first role. The 52 car tracks vary in



(a)

(b)

Fig. 11 MAP(%) for actor retrieval for the database “L’esquive”: (a) results for actor “Lydia”; (b) results of the latest STTK version for the 3 most present actors “Lydia”, “Hanane” and “Crimon” and an average on these actors.

length from 6 to 155 frames for a total of 2143 images and 30,357 SIFT vectors. The average number of SIFT chains extracted from a track is 103, varying from 3 to 283.

We use the same parameter values as for the previous actor retrieval task except for σ_2 (higher) because position prior is less relevant for cars. Indeed, if faces are meaningful mostly when captured in a frontal position, cars in movies are meaningful from any view angle. Thus, including a spatial constraint on tracked points is not relevant when view angles of the object in the tube are changing a lot.

We evaluate the performances of our system for retrieving different video objects with respect to small sizes of training set using the same data representation and the same kernel function as for actor retrieval. We train our STTK-based classifier for each car model. We focus on retrieval results for training sets of up to 8 examples picked up randomly in the whole database but preserving balance between positive and negative examples. Results are reported for the 3 car models, using MAP (Mean Average Precision) statistics on Fig. 12. These results illustrate that our retrieval system achieves to learn, to classify the 3 different classes from very few examples.

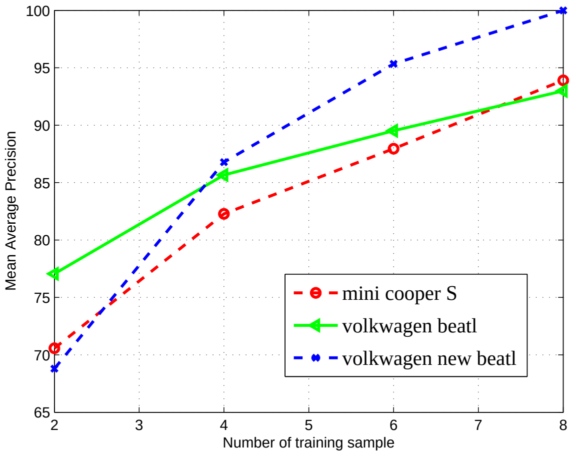


Fig. 12 MAP(%) for the car model retrieval. The learning starts with one positive and one negative examples.

5.2.3 Interactive learning car retrieval

Interactive learning [18] is a powerful active learning technique that interactively builds the training set using annotations given by the user at each iteration. One of the most popular active strategy focuses on most uncertain data [18]. In this experiment, we use this latter strategy.

Fig. 13 shows the interactive retrieval process of a car database. One car track is displayed with 4 frames in order to better illustrate the variability of car view angles. First iteration, the user initializes a query on Volkswagen Beetle (track with a green square on Fig. 13(a)), all the tubes are ranked regarding their similarity to the query, 4 tubes among the 20 first displayed contain Mini Cooper S model (Fig. 13(b)); second iteration, the user provides 3 more annotations with one more positive example(green squares) and two negative ones (red squares on Fig. 13(c)); after only 2 iterations : there is 1 tube of Mini Cooper S model remaining among the 20 first tubes displayed (Fig. 13(d)).

From our first tests on car model retrieval, we confirm the ability of our system to generalize, as it is in terms of data representation and kernel design, to other video objects rather than just faces. We also illustrate here the interest of designing kernel function as similarity measure in order to take benefit from all recent advances in machine learning such as active learning strategies. We are currently considering to implement recent boosting car detection algorithms to automatically extract more car tracks from movies, build a large database of car tracks and make it publically available.

6 Conclusions

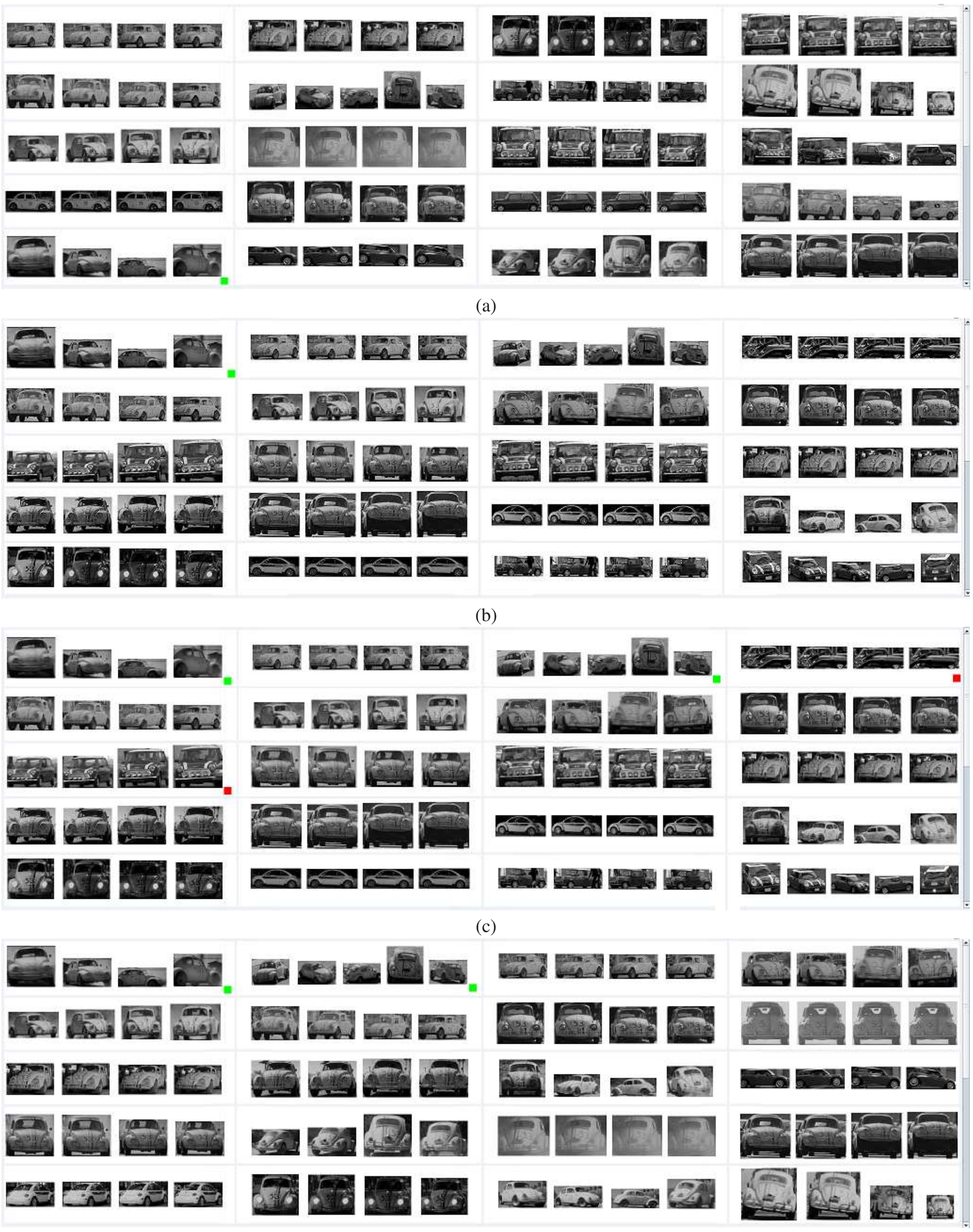
In this article, we have presented an efficient video object retrieval system, which considered video object tracks as video objects. From each video object track, a set of spatio-temporally consistent chains of tracked SIFT points is extracted. These sets are automatically filtered in order to optimize our data representation and to define our “spatio-temporal tubes”. In order to handle such complex data representation, we have designed a relevant kernel function, Spatio-Temporal Tube Kernel. We have integrated this kernel function in our multi-class SVM which provides very interesting results on databases of real movies, allowing to address both actor retrieval task and actor recognition within the same framework while dealing with unbalanced classes and occlusion. Our approach has been successfully evaluated on two real movies databases, the french movie “L’esquive” and episodes from “Buffy, the Vampire Slayer” TV series. Our method has also been tested on a car database (from real movies) and showed promising results for car retrieval task. Future work will be to embed generative models into our learning system using Bayesian kernels.

Acknowledgements We want here to thank a lot, Andrew Zisserman and Josef Sivic for providing us the data to compare our results with theirs and for the very interesting exchanges we had on this work. We also want to thank Philippe-Henri Gosselin for providing the codes of kernel-based SVM with active learning within the retrieval system RETIN.

References

1. Apostoloff, N.E., Zisserman, A.: Who are you? real-time person identification. In: BMVC (2007)
2. Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: CVPR (2009)
3. Ekenel, H.K., Stiefelhagen, R.: Why is facial occlusion a challenging problem? In: Intl. Conf. on Biometrics (ICB’09), LNCS, vol. 5558, pp. 299–308. Alghero, Italy (2009)
4. Everingham, M., Sivic, J., Zisserman, A.: Hello! my name is... buffy – automatic naming of characters in tv video. In: BMVC (2006)
5. Gosselin, P.H., Cord, M.: Active learning methods for interactive image retrieval. IEEE Trans. on Image Processing **17**(7), 1200–1211 (2008)
6. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Automatic face naming with caption-based supervision. In: CVPR, pp. 1–8 (2008)
7. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: ICCV (2007)
8. Kumar, N., Belhumeur, P., Nayar, S.K.: Face tracer: A search engine for large collections of images with faces. In: ECCV (2008)

9. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: ICIP, vol. 1, pp. I–900–I–903 vol.1 (2002)
10. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: IJCV, vol. 20, pp. 91–110 (2003)
11. Lyu, S.: Mercer kernels for object recognition with local features. In: Technical Report TR2004-520. Dartmouth College (2004)
12. Morik, K., Brockhausen, P., T., J.: Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In: ICML, pp. 268–277 (1999)
13. Osuna, E.E., Freund, R., Girosi, F.: Support vector machines: Training and applications. Tech. rep., AI Memo 1602, MIT (1997)
14. Platt, J.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. MIT Press (1998)
15. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML (2001)
16. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge, UK (2004)
17. Sivic, J., Everingham, M., Zisserman, A.: “Who are you?” – Learning person specific classifiers from video. In: CVPR (2009)
18. Tong, S., Koller, D.: Support vector machine active learning with application to text classication. JMLR **2**, 45–66 (2001)
19. Vedaldi, A.: URL http://www.vlfeat.org/ vedaldi/code/siftpp.html
20. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
21. Wallraven, C., Caputo, B., Graf, A.: Recognition with local features: the kernel recipe. In: ICCV, vol. 2, pp. 257–264 (2003)
22. Wu, G., Chang, E.: Class-boundary alignment for imbalanced dataset learning (2003)
23. Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: ICCV (2003)
24. Zhao, S., Precioso, F., Cord, M.: Spatio-temporal tube kernel for actor retrieval. In: ICIP. Cairo, Egypt (2009)
25. Zhao, S., Precioso, F., Cord, M., Philipp-Foliguet, S.: Actor retrieval system based on kernels on bags of bags. In: EUSIPCO. Lausanne, Switzerland (2008)



(a)

(b)

(c)

(d)

Fig. 13 Results of our interactive car retrieval system, one track is displayed with 4 frames: (a) Query initialization; (b) First iteration: tracks ranked regarding their similarity to the query; (c) Second iteration with one more positive example(green squares) and two negative examples (red squares); (d) Results after 2 iterations.