STTK-BASED VIDEO OBJECT RECOGNITION

Shuji Zhao, Frédéric Precioso

ETIS, CNRS, ENSEA
Univ Cergy-Pontoise, France
zhao,precioso@ensea.fr

Matthieu Cord

LIP6, CNRS
UPMC, France
matthieu.cord@lip6.fr

ABSTRACT

In this paper, we extend our video object recognition system to multi-class object recognition context, dealing with unbalanced data sets and comparing our resuls to state-of-the-art methods. Our approach is based on a Spatio-Temporal data representation, a dedicated kernel design and statistical learning techniques for object recognition. From video tracks made of segmented object regions in the successive frames, we extract sets of spatio-temporally coherent SIFT-based features, called Spatio-Temporal Tubes. To compare these complex tube objects, we integrate a Spatio-Temporal Tube Kernel (STTK) function into a multi-class classification framework with balancing process for unequal classes. Our approach is successfully evaluated on episodes from "Buffy, the Vampire Slayer" TV series which have been used in other works targeting same objectives. Our method proved to be more robust than dictionary based, facial feature based and key-frame based approaches. Our method is also tested on a small car database and preliminary results for car identification task illustrate its generalization potential.

Index Terms— Object recognition, Video object, Kernel design, multi-class.

1. INTRODUCTION

In the context of video semantic object category classification, several works have achieved interesting result especially in person recognition in movies, such as [1, 2, 3, 4]. In the framework of retrieving actors in movies, Sivic et al. [1] combines face detection using AdaBoost algorithm with an affine covariant region tracker in order to improve face extraction. Then, using a generative model (a *constellation model*), Sivic et al. exploit the underlying structure of a face to extract pose information in order to provide an invariant model of facial features. After each face instance has been represented by a 360-dimension vector based on local SIFT descriptors, the face track (a set of 360-dimension vectors) is then projected onto a dictionary of face feature examplars clusterized on a subset movie. Finally, histograms are matched with χ^2 distance. Such dictionary based methods recently focused the interest of researchers when addressing object classification in large amount of data [5].

Apostoloff and Zisserman [2] extended the previous descriptors of "face track" with 4 additional facial features and preprocessed the data before the matching process. Furthermore, the matching is not anymore based on χ^2 distance between facial feature histograms but on random-fern classifiers. In [3] Guillaumin et al. proposed an approach based on a graph of local facial features for single-person retrieval and multi-person naming. In [4] Sivic et al. introduced multiple kernel learning (MKL) based on facial features.

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 features targeting a more generic representation. In our previous work [6], the video object is represented by a "tube" of visual feature descriptors as well as spatial location of these 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, embedding a multi-class technique and balancing process, our approach is successfully evaluated on episodes from "Buffy, the Vampire Slayer" TV series and compared to state-of-the-art methods based on dictionaries [1], on random-fern classifiers [2] and key-frame based approaches. Our system is not only applied to recognition of category "person" objects, but also other category like "car model". Our approach prooves to be more robust than dictionary based, facial feature based and key-frame based approaches for actor multi-class recognition and exhibit promising generalization capability to car recognition.

2. SPATIO-TEMPORAL TUBE

2.1. Video track

We use the same face tracks as those of Apostoloff and Zisserman [2] work from Episode 2 and 5 of season 5 of TV series "Buffy, the Vampire Slayer", provided by the authors: face position, scale, frame number, with its ground truth label.

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 [7] approach in each frame of the face track. One face track is described by a set of vectors where each vector is a 128-dimensional SIFT descriptor.

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. The large variation in the size of face images in real movies can cause large variations of SIFT descriptors in face images. To solve this scaling problem, we optimize the "first octave" parameter (see [7]), regarding the scale of faces in the frames, to make it adaptive to the face image. (See [6] for details.) Hence we can extract SIFT points even if the images are small and reduce the number of irrelevant points extracted in big images, see Fig.1.

2.3. Spatio-Temporal coherent feature tracking

In order to clean up false points of interest, we then filter the SIFT points assuming the spatio-temporal coherency of relevant visual

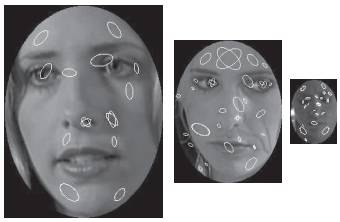
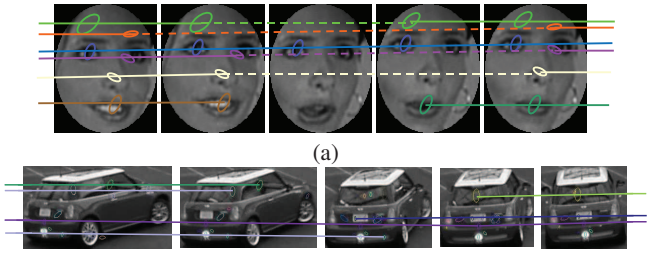


Fig. 1. Result of scale adaptive SIFT extraction. Ellipses in white represent SIFT points with scale and orientation.

features in the face track. A tracking process is used to eliminate non-persistent points in the face track. The tracking is done by selecting from two consecutive frames the 15 pairs of best matched points by feature similarity and spatial proximity and link them into chains. See solid lines of Fig.2. We then propose "intra-tube chains tracking" technique to obtain more consistent and more compact chains for each spatio-temporal tube while reducing the number of chains in a tube and thus reducing computational complexity. See dash line of Fig.2, see [6] for more details.

As a result of our spatio-temporal coherent feature tracking, a video object is represented by a tube of consistent chains of descriptors SIFT. To better represent this structural visual information, we also introduce the position of each SIFT point in the representation of points in the tube. 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". Two examples of Spatio-Temporal Tube are shown in Fig.2.



(a)

(b)

Fig. 2. Spatio-Temporal Tube, SIFT points along the same chain are of the same color (Solid lines: consistent chains, dash lines: link of two short chains). (a) a face tube; (b) a car tube.

3. STTK-BASED OBJECT RECOGNITION

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).

3.1. Spatio-Temporal Tube Kernel (STTK)

Let us denote T_i a tube, C_{ri} a chain of S_{1ri} "SIFT" vectors and S_{1ri} a 130-dimension vector (SIFT + spatial position). Using set formulation: $T_i = \{C_{1i}, \ldots, C_{ki}\}$ and $C_{ri} = \{S_{1ri}, \ldots, S_{pri}\}$. 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. We want to separate SIFT description from spatial position in the "SIFT" vector S_{1ri} in order to better handle each one of these features. This factorization and separation process is realized through two mapping functions: $\phi_f(C_{ri})$ providing the mean SIFT vector and $\phi_p(C_{ri})$ providing the mean position vector, of SIFT vectors along chain C_{ri} of tube T_i.

Our kernel function is a "power" kernel on bags weighted by the length of the chains:

$$K(T_i, T_j) = \sum_r \sum_s \frac{|C_{ri}|}{|T_i|} \frac{|C_{sj}|}{|T_j|} {k_c}^q(C_{ri}, C_{sj}), \qquad (1)$$

where $|C_{ri}|$ represents the length (number of frames) of the chain C_{ri}, $|T_i|$ represents the length of the tube T_i and $k_c(C_{ri}, C_{sj})$ is the Minor Kernel on "SIFT" Chains:

$$\begin{aligned} k_c(C_{ri}, C_{sj}) \quad = \quad & \exp\left(-\frac{1}{2\sigma_1^2}\chi^2\left(\phi_f(C_{ri}), \phi_f(C_{sj})\right)\right) \times \\ & \exp\left(-\frac{(\mathbf{x}_{ri} - \mathbf{x}_{sj})^2 + \left(\mathbf{y}_{ri} - \mathbf{y}_{sj}\right)^2}{2\sigma_2^2}\right) \qquad (2) \end{aligned}$$

where the first term is the feature kernel between the averaged SIFT vectors of two chains and the second term is the position kernel between the averaged positions vectors $(\mathbf{x}_{ri}, \mathbf{y}_{ri}) = \phi_p(C_{ri})$ of the same two chains. Let us remind that the position of a SIFT point (x, y) is normalized by the size of the face image. The position term introduces the importance of the comparison between two chains approximately at the same position in the object detected area.

One second effect of the kernel of Eq.1 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. One technique is then to normalize the kernel as described in Chapter 3 of [8]:

$$K'(T_i, T_j) = \frac{K(T_i, T_j)}{\sqrt{K(T_i, T_i) \cdot K(T_j, T_j)}} \qquad (3)$$

After normalization of kernel, the similarity of each tube with itself is always 1, however high its "size".

In Chapter 3 of [8], one can find all the kernel design properties which are involved in our own kernel design and thus proove that K as well as K' are valid kernel functions.

3.2. Multi-class SVM for actor recognition

We use kernel-based SVM as machine learning classification technique: the SVM is a robust and powerful classification technique for two-class problems. For the multi-class classification task, we use the one-vs-all approach, similar to [9], where N binary SVMs are trained for solving a N-class problem : Labels are assigned to data according to highest relevance among the N SVMs.

One of the problems of the multi-class SVM classification is often to deal with unbalanced datasets where negative examples far outnumber positive examples.

Most of SVM classifiers use the SMO algorithm of [10] where the classification function is computed by resolving the well-known

QP problem with the KKT (Karush-Kuhn-Tucker) conditions. With the KKT conditions, all the negative samples (majority class) are situated near the negative support hyperplane while the positive samples (minority class) are near the separation boundary. As a consequence, a query falling between the positive and negative Support Vectors is likely to be classified as a negative one.

In our experimentation we use the Biased Penalties proposed by Morik et al. [11], which improves the precision/recall result for object recognition.

4. EXPERIMENTS

We have tested our actor recognition framework on database "Buffy" as in [2] with same tracks ground truth data to compare the performance of our actor recognition framework with these previous works. From these face tracks segmented, 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 our machine learning system, with SVM core using our STTK kernel. We have also tested our method of balancing unequal classes, compare our STTK based system with a simple key-frame based approach and dictionary based approach on "Buffy" database. Our system is finally tested on another video object category, car model, and obtains interesting results.

4.1. 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 [2]: the first is the intra-episode recognition, we train our classifiers on the 159 train 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 [2]); 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 machine learning system using our STTK kernel functions of Eq.1, 2 and 3. In our work, we set $q = 2, \sigma_1 = 3, \sigma_2 = \sqrt{0.05}$. Fig.3 shows a sample set of identification results of our STTK based actor recognition system.

Fig. 3. Sample identifications from episode 05-05. Green squares mean correctly matched faces, while red squares mean failure cases.

4.1.1. Comparison with key-frame based approach

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-base approach replacing two chains C_{ri} and C_{sj} by two SIFT vectors S_{ri} and S_{sj}. See Fig.4(a) 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.

4.1.2. Comparison with Random-ferns based approach

We compare our approach with the Random-ferns based approach of Apostoloff and Zisserman work [2] for intra-episode and inter-episode cases. Regarding the precision/recall curves of [2], our approach performs better or at least as well as Table1 shows it. Precision of Random-ferns approach have been extracted for several Recall rate from the curves of [2] as fairly as possible.

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

Table 1. Quantitative precision results at different levels of recall.

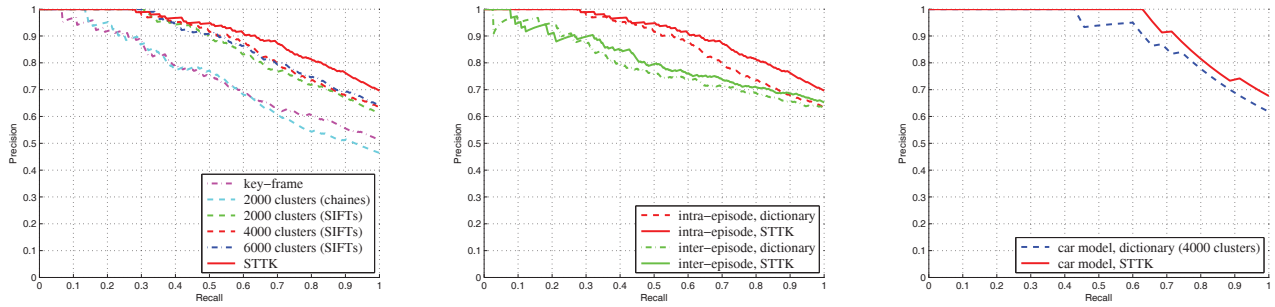
4.1.3. Comparison with dictionary based approach

To compare our STTK based approach to dictionary-based approaches, we consider both intra-episode and inter-episode test scenarii on "Buffy" database.

We firstly test intra-episode scenario to find the best parameters for the dictionary-based approach. Using the standard k-means algorithm on the episode 2 train database, we obtain clusters and build a dictionary based on theses clusters. Each face track tube is then represented by a histogram of "visual words" and is put into a SVM classifier with same multi-classes framework and balancing strategy as in the STTK based approach. We train our classifiers on 213,408 SIFTs extracted from the 159 train tracks of episode 2 season 5 by two steps: firstly we clusterize these SIFT vectors by k-means to obtain n prototypes, then we compute a n bins histogram for each tube and train a SVM for each class. In our experiment, we considered 2000, 4000 and 6000 clusters respectively for the dictionary size. We then evaulated the retrieval performances on the 374 test tracks of episode 2 season 5. The kernel we used is Gaussian L^2 kernel with $\sigma = 1$ (the best parameters for the best kernel in our experiment). As showed in Fig.4(a), 4000 clusters provide the best results among dictionary-based classifications.

We also evaluated the dictionary-based approach on our pre-extracted spatio-temporal chains in each tube, instead of considering the simple SIFTs vectors. As the number of chains in the training set (12,883 chains) is significantly reduced compared to the number of SIFT vectors, we tested smaller sizes of dictionary, and found 2000 as the optimal number of clusters. Because of this small number of chains for building a dictionary, the precision/recall curve is below the simple SIFTs vector dictionaries, see Fig.4(a).

In inter-episode case, we train our classifiers in the same way as for intra-episode on 285,053 SIFTs extracted from the 533 tracks (one from every two frames to reduce the amount SIFT vectors) of episode 2 season 5, then we test on the 482 test tracks of episode



(a) (b) (c)

Fig. 4. 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) comparing dictionary based approach with different numbers of clusters and STTK based approach; (b) comparing dictionary/STTK based approach in intra-episode and inter-episode; (c) car model recognition.

5 season 5. See Fig.4(b) for the comparison of dictionary-based and STTK-based SVMs on intra-episode and on inter-episode data. The STTK-based system is far better than the dictionary-based approach, especially in the high recall phase, illustrating that the proposed spatio-temporal structure of SIFT chains matched through a dedicated kernel function handles more information than the classic dictionary-based approach.

4.2. Car model recognition in real movies

We test our STTK-based object recognition system on a database of car tracks containing 3 car model (volcano beatle, mini austin and mini cooper S), extracted by hand, from different movies "The Italian Job" (both versions 1969 and 2002) and "Herbie: Fully Loaded" (2004). The movies we chose to extract enough car tracks are movies in which cars get some kind of first role. The 52 car tracks vary in 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 158. See on Fig.5 three significant examples of car video tubes illustrating high variations of pose, size and conditions.

Fig. 5. Samples of car model tubes.

From our very first tests on car model recognition (see Fig.4(c) for the precision/recall curves), using the 52 tubes, 17 for training and 35 for test, we confirm the ability of our system to generalize, as it is, to other multi-class object recognition. Indeed, all the parameters have been kept identical between face and car experiments. We again did a comparison with dictionary-based approaches and present the result for the best size we have tested of 4000 clusters.

We are currently building a larger car database and we will make it publically available in order to allow thus anyone to test either car detection algorithms (several boosting-based algorithms have recently been proposed) or object recognition systems as we are considering here.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a multi-class kernel-based object recognition system. We have thoroughly compared our approach to state-of-the-art methods, keyframe-based, dictrionary-base and random-ferns approaches, showing that our method provides very interesting results on reference databases from real movies. The proposed system is robust to unbalanced class issues. Our method also tested on a car database, with exactly same parameters as face recognition experiments, shows promising preliminary results for car recognition task and thus gives insight into its generalization potential. Future work will be to embed generative models into our learning system using Bayesian kernels in order to intend getting rid off object detection phase. We will also make publically available the car database with its ground truth we are working on.

6. REFERENCES

[1] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: video shot retrieval for face sets," in *CIVR*, Singapore, 2005.

[2] N. E. Apostoloff and A. Zisserman, "Who are you? real-time person identification," in *BMVC*, 2007.

[3] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Automatic face naming with caption-based supervision," in *CVPR*, 2008.

[4] J. Sivic, M. Everingham, and A. Zisserman, ""Who are you?" – Learning person specific classifiers from video," in *CVPR*, 2009.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[6] S. Zhao, F. Precioso, and M. Cord, "Spatio-temporal tube kernel for actor retrieval," in *ICIP*, Cairo, Egypt, November 2009.

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *IJCV*, 2003, vol. 20, pp. 91–110.

[8] John Shawe-Taylor and Nello Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, June 2004.

[9] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: Component-based versus global approaches," *CVIU*, 2003.

[10] J. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, MIT Press, April 1998.

[11] P. Morik, K. Brockhausen and Joachims T., "Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring," in *ICML*, 1999, pp. 268–277.