Semantic Concept Detection Using Dense Codeword Motion

Claudiu Tănase and Bernard Mérialdo

EURECOM
Campus SophiaTech
450 Route des Chappes
06410 Biot France

Abstract. When detecting semantic concepts in video, much of the existing research in content-based classification uses keyframe information only. Particularly the combination between local features such as SIFT and the Bag of Words model is very popular with TRECVID participants. The few existing motion and spatiotemporal descriptors are computationally heavy and become impractical when applied on large datasets such as TRECVID. In this paper, we propose a way to efficiently combine positional motion obtained from optic flow in the keyframe with information given by the Dense SIFT Bag of Words feature. The features we propose work by spatially binning motion vectors belonging to the same codeword into separate histograms describing movement direction (left, right, vertical, zero, etc.). Classifiers are mapped using the homogeneous kernel map techinque for approximating the $\chi 2$ kernel and then trained efficiently using linear SVM. By using a simple linear fusion technique we can improve the Mean Average Precision of the Bag of Words DSIFT classifier on the TRECVID 2010 Semantic Indexing benchmark from 0.0924 to 0.0972, which is confirmed to be a statistically significant increase based on standardized TRECVID randomization tests.

Keywords: content based video retrieval, semantic indexing, TRECVID, spatio-temporal features, motion feature.

1 Introduction

With the ever increasing accessibility of devices capable of recording video and the popularity of video hosting websites, large collections of user submitted videos are becoming the focus of important research in content-based multimedia retrieval and classification. The core problem of automatically categorizing a new video based on its content has proven to be considerably harder than the image counterpart. The goal of our research is to simply be able to tell whether a predefined semantic concept such as "car" or "running" is present or not in a video.

Most of the state on the art in video concept detection works almost exclusively with image features by extracting a relevant keyframe from the video. A very successful combination, found in almost every submission to the Semantic

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-319-02895-8_64

J. Blanc-Talon et al. (Eds.): ACIVS 2013, LNCS 8192, pp. 705–713, 2013.

Indexing challenge of TRECVID[10], is the SIFT descriptor and Bag of Words model.[9] We believe that the perceived motion of each of the SIFT patterns is useful in recognition. In this work we are upgrading the BoW representation of the Dense SIFT descriptor with motion information extracted locally from the corresponding keypoint positions in the frame. Our feature is based on the idea that because SIFT patches successfully describe object patterns, the objects' motion in the scene is in some measure captured by the motion of the SIFT codewords[14]. The strength of our method is in the fact that it can reuse the information stored in the DSIFT Bag of Words feature vectors, thus greatly reducing feature extraction time. Classification takes in average around one second for a feature matrix of 119685 vectors of dimensionality 2500.

In this paper we propose a new set of content description features, derived from DSIFT, that take into account the motion of the SIFT patches. Using a simple binning technique, we create 3 features named ZN, ZHV and ZLRUD that capture not only the codeword information stored in DSIFT histograms but also the quantity and direction of movement of the SIFT patch. The addition of our features to an existing concept detection system based on DSIFT features comes with the relatively low cost of extracting sparse optic flow from one keyframe per video shot. By using the well-known homogeneous kernel map[13] method we can efficiently approximate the non-linear $\chi 2$ kernel and train linear SVMs in a fraction of the non-linear SVM computation time. By using a linear score combination, we combine DSIFT with our Z features and obtain an increase in Mean Average Precision (MAP) of about 5%. By applying an official TRECVID tool that compares submission runs based on randomization testing, we are able to confirm that the aforementioned improvement is statistically significant.

2 Related Work

One particular web video collection has been the subject of considerable research in recent years. In the TRECVID[10] Semantic Indexing task, a benchmark of annotated videos is used for detecting a large set of predefined concepts. The traditional concept detection in video, as shown by works published in TRECVID workshops, uses keyframe techniques. One keyframe is selected from the shot in question and all subsequent processing deals with the keyframe as sole representative of the shot, much like a CBIR system. Compared to the few existing spatio-temporal content descriptors[8,1], this approach is hugely more efficient in time and memory. The disadvantage is that obviously all the motion and sequence information is lost.

Of these keyframe methods, the Bag of Words (BoW) technique has been prevailing in TRECVID for many years. Although newer methods like Fisher vectors [6] and super-vectors [17] supersede BoW, it remains widely used in the community. In the BoW model a set of local visual features is extracted from the image. As a result of K-means clustering of a pool of features, a codebook a.k.a. visual dictionary is created. Each centroid obtained in the clustering represents a codeword or visual word. According to the precomputed codebook each extracted

feature is assigned to its nearest element in the codebook. The final feature is a histogram of occurrences of each codeword. In the vast majority of situations, this technique uses SIFT[9] as the visual feature, whether using the original interest point detection or by dense sampling[12]. Recent work seems to suggest that in this context the dense extraction seems to slightly outperform interest point methods[4,7,16].

There are several local features called spatio-temporal descriptors that describe image sequences, most notably used in the action recognition community. One notable example is Laptev's Spatio-Temporal Interest Point descriptor[8] (STIP), which detects 3D interest points using a 3D extension of the Harris operator and describes them using histograms of oriented gradients and histograms of flow (HOGHOF). Wang's dense trajectories descriptor[15] extracts and tracks features throughout the entire video volume. Chen's MoSIFT[1] is an extension of SIFT that adds in a similar feature where the gradient is replaced by optic flow. Since all these methods analyze a 3D volume instead of a 2D image, the computational complexity is much higher than the standard keyframe approaches, to the point that implementing a spatio-temporal technique on TRECVID might prove too computationally heavy for some systems. By comparison, our feature computes the optic flow on a single keyframe in the video. The extraction of DSIFT, codebook assignment and motion feature construction take in average 8.27 seconds for any TRECVID video, while any of the 3 descriptors mentioned earlier take in average well beyond 30 seconds to compute, depending on video length.

Our approach shares some similarity with part of the work of Wang et. al.[14] in that codeword motion is considered. Two important aspects differ. Firstly, the input features come from dense sampling in our work and from keypoint detections in [14]. Features computed at keypoints will extract information from salient zones in the image and will ignore uniform or weakly-textured zones. They concentrate on details so that they are better suitable at object recognition rather than detection. Dense sampling ensures that every pixel in an image is covered by at least one patch, which makes it less likely to miss an object. This leads to the background forming an important element. Also, in dense sampling more features are selected and variability is higher. In short, keypoints are better for precision, dense is better for recall. The second difference is in the way motionless patches are processed. In [14] an orientation histogram is built using projections that cumulate in each bin. Patches with small amounts of motion contribute little to the resulting orientation histogram. Our features contain a Z (zero) bin, which stores the number of patches with little or no motion, which means that the information on static codewords is not lost.

3 Extracting Codeword Motion

Dense SIFT works by extracting features from evenly spaced keypoints. In our version we use a grid of size 8 pixels. The densely extracted SIFT features are quantized and assigned to one of the $k = 500$ codewords. The value of k has been

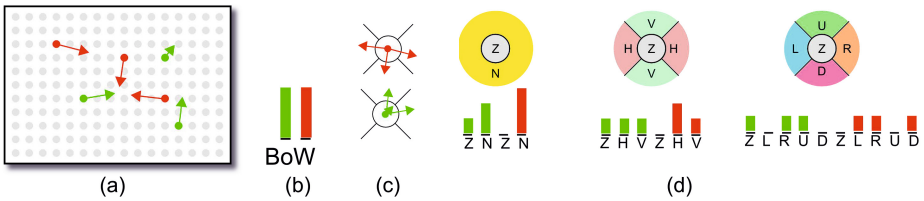

Fig. 1. Overview of feature construction: (a) DSIFT patches assigned to codewords (green and red) are extracted along with their oprtic flow. (b) Bag of Words histogram is built by counting the occurrences of each codeword. (c) Motion vectors are grouped by codeword. (d) Histograms are being constructed for every bin and every codeword by counting the number of flow vectors in each bin.

empirically chosen as a good compromise between performance and computation speed. In the normal DSIFT, keypoint locations are ignored and only the total count of codeword occurrences are taken into consideration. However we keep for each keypoint position x, y the index of the codeword c.

In order to extract motion, we first access the keyframe in the video file. We then advance by a small time interval and extract a second frame corresponding to slightly later time in order to have sufficient difference. The motion between these frames is subjected to a mostly uniform background camera movement that can be compensated for. We use a camera stabilization function similar to the one in [5]. This method does dominant motion compensation by estimating a homography with RANSAC over detected feature correspondences. This homography is then used to produce a synthetic motion vector field modeling the camera movement which is used as an initial estimate for the full-frame optic flow using Farneback's method[3]. The displacement between the synthetic background motion field and the actual motion field can then be used as an estimation of foreground objects, since motion in background areas is compensated for. Having computed the motion compensated flow, we can now sample it at the keypoint positions x, y. Thus, for every keypoint i we now have its coordinates x_i, y_i, a codeword value c_i and the optic flow f_i^x, f_i^y.

4 Spatial Codeword Motion Histograms

We now group our features by codeword. For each codeword c, we quantize the flow coordinates f_i^x, f_i^y according to the spatial histograms in figure 1. The bin corresponding to the region where the (f_i^x, f_i^y) point falls is incremented. The zero (Z) bin will capture features with zero or small motion. The value of the Z bin radius θ has been chosen as the median of all the optic flow velocities in the collection in order to ensure the balance between the number of features falling inside and outside of Z (which is the non-zero bin N). Since there is an intuitive conceptual distinction between horizontal and vertical movement, we separate our space in 2 corresponding bins. Horizontal and vertical bins H and

V take advantage of origin symmetry, are spatially discontinuous and quantize feature orientation. Left, right, up and down bins L, R, U and D separate motion direction (bearing). The 3 features we are studying in this paper are:

1. ZN which describes *whether the codeword moves or not*
2. ZHV which discriminates between codewords moving horizontally and vertically, thus contains information on *orientation*, and
3. ZLRUD which discriminates codewords moving left, right, up and down, therefore encodes the *direction*

Since there are several spatial bins for each codeword, the final feature size will have size $2 \times K$ for the ZN variant, $3 \times K$ for ZHV and $5 \times K$ for ZLRUD. Since Z features are decompositions of the DSIFT features, the relation between these bins is given by the following formula:

$$DSIFT = Z + N = Z + H + V = Z + L + R + U + D \tag{1}$$

The baseline DSIFT BoW approach works by directly counting codeword occurrences, which makes it equivalent to a single bin covering all the space. All resulting histograms are normalized using the L1 norm.

5 Classification and Fusion

The experimental setup closely follows the TRECVID 2010 Semantic Indexing evaluation. We are evaluating 50 concepts, with sparse training annotations available on a development set containing 119,685 sequences, and applied on a test set of 146,788 sequences. We also use the MAP (Mean Average Precision) as performance measure. In order to benefit from the superior classification power of non-linear kernels, we employ kernel approximation techniques described in[13]. In practice, the $\chi 2$ kernel seems to perform very well when using Bag of Words features. We map our features using the homogeneous kernel map of order $N = 3$, implemented in the Scikit-learn library[11] which yields a new feature dimensionality of $7\times$ the original one. The new feature vectors can be used to train a linear SVM, which will approximate the non-linear $\chi 2$ SVM classifier. For that we use the Liblinear[2] implementation found in Scikit-learn by training on half of the development set (59,842 sequences) and cross-validating on the other half in order to optimize the C parameter of the SVM. After finding the optimal value of C, we train another linear SVM with this value of C on the entire development set and test on the testing set. The classifier confidence values found in the validation set are kept for later use in the linear fusion. We apply this procedure for DSIFT, ZN, ZHV and ZLRUD features. As it is routinely done in TRECVID, this process is done using training annotations from one of the 50 concepts at a time. Using the SVM confidence values and the ground truth on the testing set we compute the Average Precision for each concept. The run is finally evaluated by averaging these average precisions (MAP).

Classification scores from the different features are then combined using late fusion. This is done by finding the linear combination of score weights that

maximize the MAP on the validation set and reapply these weights on the testing set confidence values. Since the computation of weighted sums and of the MAP are almost instantaneous, a grid search on the weight values is possible. We experiment with the fusion of the baseline DSIFT, ZN, ZHV and ZLRUD, as described in the next section. Each weight is tested in 0.1 increments.

We use the TRECVID randomization testing[10] to estimate the statistical significance of the increase in MAP. Each test implements a partial randomization test of the hypothesis that two search runs, whose effectiveness is measured by MAP, are significantly different - against the null hypothesis that the differences are due to chance. We use this approach to pairwise compare DSIFT, ZN, ZHV, ZLRUD and the fusion result.

6 Experimental Results

Table 1 shows the MAP for the classifier DSIFT, ZN, ZHV and ZLRUD. Although more information is contained within the Z features than in DSIFT, their overall MAP is lower. The reason is that the 4 features have different dimensionalities (DSIFT=500, ZN=1000, ZHV=1500, ZLRUD=2500) and are trained with the same classification technique. A more robust comparison would have been for instance DSIFT with a codebook of size $k = 2500$ compared to the present ZLRUD feature, but such a comparison would require calculating DSIFT features and Bag of Words for both k=500 and k=2500, which would defeat the purpose of this work.

Table 1. Mean average precision of the 4 features and fusion

	DSIFT	ZN	ZHV	ZLRUD	fusion
dim	500	1,000	1,500	2,500	n/a
MAP	0.0924	0.0853	0.0809	0.0723	0.0972

Figure 2 shows the weight of each feature in fusion and can be interpreted as an indication of what type of movement is the most informative for classifying the concept, e.g. if the DSIFT component has a high weight, then the concept is more easily classified based only on static visual information. High ZN weight means that the presence or absence of movement is a good cue for detecting the concept. ZHV has high weight if the direction of movement is important and ZLRUD is high when both direction and orientation of movement are relevant.

The TRECVID randomized test for statistical significance have confirmed that for a significance level of 0.05 the fusion run statistically outperforms all of the features. The conclusion of said test is that the improvement of the MAP from 0.0924 to 0.0972 is in fact a statistically significant improvement and not due to chance.

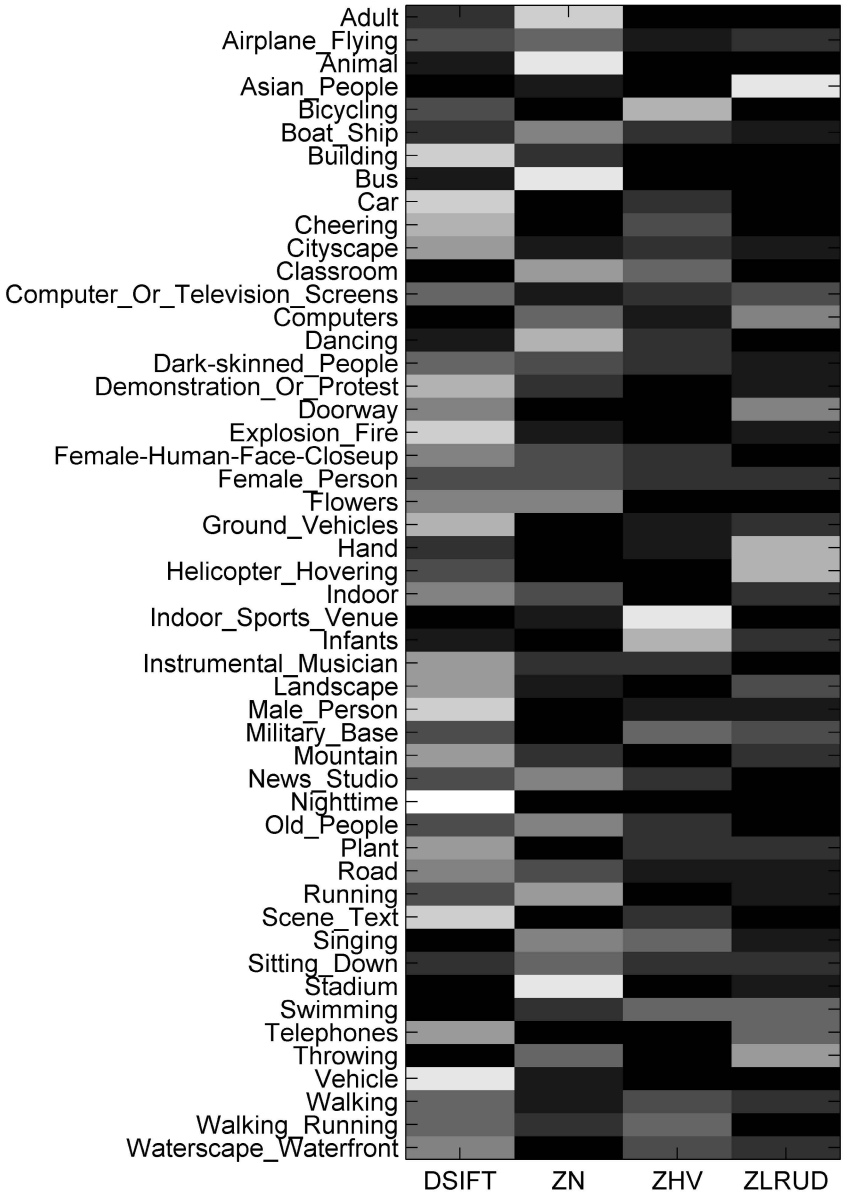

Fig. 2. Weight contributions of each feature in the optimal best performing concept classifier. Values are grayscale, white is one, black is zero.

7 Conclusions

In this paper we have presented a new feature that builds on the DSIFT Bag of Words classifier by incorporating local motion information. The proposed features are constructed using optic flow information at the DSIFT patch positions and the corresponding codewords. Using minimal computation, 3 spatial histogram features ZN, ZHV and ZLRUD are constructed from existing DSIFT feature codeword data and optic flow information and are mapped using the homogeneous kernel maps and classified using linear SVM. The strength of the method is that it relies on the widely available DSIFT bag of words feature data, and requires minimal feature extraction, namely optic flow at a keyframe level, and that linear classification is extremely fast, thanks to the homogeneous kernel map. Linear descriptor fusion show that the new features can improve the performance of the retrieval system by a statistically significant 5% without requiring any of the more computationally complex spatio-temporal techniques.

References

1. Chen, M., Hauptmann, A.: Mosift: Recognizing human actions in surveillance videos (2009)
2. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
3. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003)
4. Gorisse, D., Precioso, F.: IRIM at TRECVID 2010: Semantic Indexing and Instance Search. In: TREC Online Proceedings, Gaithersburg, United States. gDR ISIS (November 2010)
5. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 494–507. Springer, Heidelberg (2010)
6. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011), `http://hal.inria.fr/inria-00633013`
7. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, vol. 1, pp. 604–610. IEEE (2005)
8. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings of the Ninth IEEE International Conference on Computer Vision 2003, vol. 1, pp. 432–439 (October 2003)
9. Lowe, D.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision 1999, vol. 2, pp. 1150–1157. IEEE (1999)
10. Over, P., Awad, G., Fiscus, J., Antonishek, B., Michel, M., Smeaton, A., Kraaij, W., Quénot, G., et al.: An overview of the goals, tasks, data, evaluation mechanisms and metrics. In: TRECVID 2011-TREC Video Retrieval Evaluation Online (2011)

11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research 12, 2825–2830 (2011)
12. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008)
13. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3539–3546. IEEE ((2010)
14. Wang, F., Jiang, Y.G., Ngo, C.W.: Video event detection using motion relativity and visual relatedness. In: Proceedings of the 16th ACM International Conference on Multimedia, pp. 239–248. ACM (2008)
15. Wang, H., Klaser, A., Schmid, C., Liu, C.: Action recognition by dense trajectories. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3169–3176. IEEE (2011)
16. Wang, H., Ullah, M., Klaser, A., Laptev, I., Schmid, C., et al.: Evaluation of local spatio-temporal features for action recognition. In: BMVC 2009-British Machine Vision Conference (2009)
17. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)