JOANNEUM RESEARCH and Vienna University of Technology at TRECVID 2012: Semantic Indexing and Instance Search

Werner Bailer*, Robert Sorschag†, Harald Stiegler*, Christoph Wiedner*
*JOANNEUM RESEARCH, DIGITAL – Institute for Information and Communication Technologies
8010 Graz, Austria
Email: {firstName.lastName@joanneum.at}
†Vienna University of Technology, Interactive Media Systems Group
1040 Vienna, Austria
Email: sorschag@ims.tuwien.ac.at

ABSTRACT

We participated in two tasks: semantic indexing (SIN) and instance search (INS).

SIN runs

We submitted one light run, using multiple kernel learning (MKL) to combine longest commmon subsequence kernels with different similarity parameters. The features are SIFT bag of features histograms and global color and texture features.

The performance for some concepts is in the expected range, while the infAP score is extremely low for five of the concepts. This issue is not observed when applying the same approach to the 2011 data, and needs further investigation.

INS runs

We applied two approaches with quite complementary properties: One with preprocessing and indexing (based on a bag-of-features (BoF) approach using Color-SIFT), and very fast query times (at most one minute), and without any preprocessing, but performing SIFT extraction and matching at query time. We submitted the following four runs:

- JRSVUT1: indexed Color-SIFT
- JRSVUT2: SIFT matching at query
- JRSVUT3: top results of SIFT matching at query, and indexed Color-SIFT (densely sampled) results
- JRSVUT4: top results of SIFT matching at query, and indexed Color-SIFT (extracted from DoG points) results

The indexing method is very fast, but results are poor. The SIFT matching at query time provides good results, at or close to the best for some queries. Queries with small sample images or no distinctive visual properties yield very low performance. Fusion improves results for many queries, but removes a large number of correct hits for a few queries.

I. SEMANTIC INDEXING

For the semantic indexing task we use a set of low-level features extracted from key frames and train a classifier for each concept using SVMs. We use multiple kernel learning (MKL) to combine different parameterizations of a sequence-based kernel. In the following, we briefly describe the features and the kernels we used in the experiments. We then discuss our results.

A. Features

1) MPEG-7: The following MPEG-7 [1] image features were extracted globally:

Color Layout describes the spatial distribution of colors. This feature is computed by clustering the image into 8x8 blocks and deriving the average value for each block. After computation of DCT and encoding, a set of low frequency DCT components is selected (6 for the Y, 3 for the Cb and Cr plane).

Dominant Color consists of a small number of representative colors, the fraction of the image represented by each color cluster and its variance. We use three dominant colors extracted by mean shift color clustering [2].

Color Structure captures both, color content and information about the spatial arrangement of the colors. Specifically, we compute a 32-bin histogram that counts the number of times a color is present in an 8x8 windowed neighborhood, as this window progresses over the image rows and columns.

EdgeHistogram represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. We use a global histogram generated directly from the local edge histograms of 4×4 sub-images.

2) Bag-of-features (BoF): About 300 densely sampled image regions from 3 different scales are selected per key frame. A 384 dimensional Color-SIFT descriptor (4×4 subregions, 8 directions for orientation histograms, separately computed from all 3 RGB color channels) is extracted for each of these regions without computation of a dominant orientation. The Color-SIFT approach was motivated by the work of [3] that show an increased performance of the BoF-SIFT approach when additional color information was added. Thus, we extracted the Color-SIFT descriptors in a similar way.

Higher-level features are then generated by the popular bag-of-features (BoF) approach, where the Color-SIFT features are mapped to codewords. These codewords are generated in an offline step using the k-means algorithm on about 100,000 features from randomly selected Flickr images. We use codebooks with 100 codewords which leads to two 100 dimensional BoF features for each key frame. Each entry in one of these BoF features states the number of times a specific codeword was detected in a key frame. The mapping between Color-SIFT features from a key frame and their codewords is identified by nearest neighbor search with Euclidean distance. Beside global BoFs of the entire key frames, we generated further versions where the key frames are split into 2×2, 1×3, 3×1, and 3×3 regions in horizontal and vertical direction. A 100 dimensional BoF feature is then generated for each partition and they are concatenated to 300, 400, and 900 dimensional features.

B. Multiple kernel learning with sequence-based kernels

Kernel methods, most notably Support Vector Machines (SVMs), have been widely applied to classification problems, also due to the availability of toolkits such as LibSVM [4]. SVM based classifiers are also commonly used for concept classification based on visual features. Sequence-based kernels, i.e., kernel functions that are able to determine the similarity of sequences of feature vectors, are one of the methods proposed for capturing the temporal dimension of dynamic concepts. Experiments have shown (see e.g. [5]) that concept classifiers using sequence-based kernels outperform those using kernels matching only the individual feature vectors of the samples of a segment independently.

The general approach of sequence-based kernels is to define a kernel function on a sequence[1] of feature vectors from two video segments (which may be regularly or irregularly sampled). Elements in the sequence represent the feature vectors of individual frames, and a base distance/similarity function (which can be a kernel itself) is applied to them. Then the kernel value for the two sequences is determined from the base distance similarity value, e.g., by choosing some optimal alignment, a weighted combination of different alignments etc. The latter step includes many properties that discriminate the different types of sequence-based kernels, such as thresholds for the base distance/similarity, constraints on gaps in the alignment, etc.

Existing work on sequence-based kernels either uses a single type of feature (e.g., bag of visual words) or combines the feature vectors of frames (e.g., by a weighted sum or product). When using multiple features, the optimal alignments between two sequences can vary in the different types of features. For example, for kernels supporting gaps in the alignment, a strong short-term lighting change might cause a gap in the alignment of a color feature, while a continuous alignment may still be possible for a texture based feature, thus increasing the value of the kernel function over the case where a gap is introduced for all features together. Also, audio and visual features may be extracted with different temporal sampling rates, so that they cannot be easily combined into feature vectors for a certain time point.

The optimal alignment determined by a sequence-based kernel also depends on parameters such as a similarity threshold for the values of the kernel function between individual elements in the sequence, the tolerable gap, or whether to base optimality on the length of the match or the mean similarity of the matching elements. Depending on the choice of these parameters, different alignments with associated values of the kernel function are possible, and it is often not possible to tell which of the alignments is "correct" or just "better" for a certain task. As it is difficult to determine the weights for the different alignments, they are often based on the same optimality criteria as in kernels choosing a single alignment, e.g. length of the matching sequence or they weighted equally.

Multiple kernel learning (MKL) has been proposed for problems, where instead of choosing a kernel a priori, weights for combining different kernels are learned together with the model [6]. In this paper, we apply MKL for combining different sequence-based kernels for video concept detection, with different parameterizations and using different features, as well as for combining sequence-based kernels with kernels treating the samples independently.

Kernels based on the longest common subsequence (LCSS) algorithm have been proposed in [5], [7]. The kernel described in [5] allows plugging in any kernel for measuring the distance between the feature vectors of the samples of the two sequences, and includes the similarities in the result of the kernel. The kernel uses a recursive definition of LCSS and a threshold θ_{sim} to decide if two feature vectors are considered as matching.

$$\text{LCSS}(X, Y, \theta_{\text{sim}}) = \begin{cases} 0, & \text{if } |X| = 0 \vee |Y| = 0, \\ \kappa_f(x_{|X|}, y_{|Y|}) + & \\ \quad \text{LCSS}(\text{Head}(X), \text{Head}(Y)), & \text{if } \kappa_f(x_{|X|}, y_{|Y|}) \geq \theta_{\text{sim}}, \\ \max(\text{LCSS}(\text{Head}(X), Y), & \\ \quad \text{LCSS}(X, \text{Head}(Y))) & \text{otherwise,} \end{cases} \tag{1}$$

where κ_f is a specific kernel for feature f, θ_{sim} is a threshold to consider two feature vectors as matching and $\text{Head}(X) = (x_1, \ldots, x_{|X|-1})$.

Multiple kernel learning (MKL) is an approach that considers a set of kernels potentially appropriate for the respective problem, and estimates both the parameters of the individual kernels as well as their relative weights during the training phase. In particular, we discuss $L1$-norm MKL, which defines the kernel to be learned as a linearly weighted sum of different kernel functions. The authors of [8] discuss how this approach can not only be applied to combining different types of kernels, but also to combining a set of instances of kernels with different parameters, where the parameters can include

[1] In this paper, the term sequence denotes a possibly non-contiguous subsequence.

features, parameters for feature extraction, kernel parameters, etc. The parameter space can thus be potentially infinite.

Let $\mathcal{X}$ denote a set of feature sequences $(X^1, \ldots, X^F)$, describing the same video segment with different features.

We can choose sets of parameters $\theta = (f, \theta_{\text{sim}})$ from the parameter space Θ and can define the combined kernel as weighted sum of instances of the unified kernel with these parameters as

$$\kappa^*(\mathcal{X}, \mathcal{Y}) = \sum_{\theta \in \Theta} \beta_\theta \kappa(X^f, Y^f, \theta_{\text{sim}}) \tag{2}$$

where β_θ is the weight of a specific parameter set θ (i.e., the subkernel weight of the respective kernel instance), with $\beta_\theta \geq 0, \sum_{\beta_\theta} = 1, \forall \theta \in \Theta$.

In order to apply sequence-based kernels, we have sampled more key frames based on visual activity than provided in the TRECVID master shot reference. For the MPEG-7 features we use the kernel proposed in [9], and for the bags of visual words we use the histogram intersection kernel [10]. For concepts that have a very high number of positive samples, the number of samples has been limited to the key frames of 1,000 shots (randomly sampled), and balanced with the same number of randomly selected negative samples. For solving the MKL problem we use the Shogun framework [11], using the interleaved optimization method described in [12].

We parameterize 35 subkernels of the MKL problem. The 35 parameter sets contain the seven features described above (MPEG-7 Color Layout and Edge Histogram, and Color-SIFT BoF histograms over five spatial configurations), each using the LCS kernel with 5 different values of the similarity threshold $\theta_{\text{sim}} \in \{0.10, 0.30, 0.50, 0.70, 0.90\}$. On the training data, we learn the models for each of the kernels as well as the relative weights of the subkernels. We have not used the global features Dominant Color and Color Structure, as we found that the resulting similarity matrices have a significantly higher fraction of negative eigenvalues than for other features when using the longest common subsequence kernel, i.e., they are not very discriminative in determining a good alignment. The fractions of negative eigenvalues for different features are shown in Figure 1.

C. Results

The performance of the SIN run are visualized in Figure 2. The inferred average precision is rather low, and generally below the median of all runs. The results of MKL training include the weights of the different kernels with different parameters and applied to different features. The fractions of the weights are shown in Figure 3 (features) and Figure 4 (thresholds) respectively. For queries where the global features are dominant, the performance is very low. However, the inverse does not hold in all cases.

D. Conclusion

While the MKL approach has shown to outperform kernels with specific parameters and a fixed combination of features on the TRECVID SIN 2011 data set, the results for 2012 are not as good as expected. While the results are comparable to 2011 for a some concepts, the performance is extremely low for five of the concepts. This issue needs further investigation.

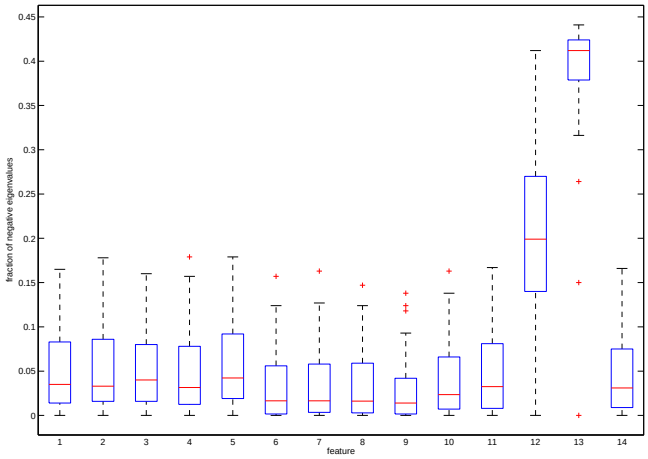

Fig. 1. Fraction of negative eigenvalues for different features. BoF Color Layout (CL): (1) CL 1x3, (2) CL 2x2, (3) CL 3x1, (4) CL 3x3, (5) CL 1x1; BoF SIFT: (6) SIFT 1x3, (7) SIFT 2x2, (8) SIFT 3x1, (9) SIFT 3x3, (10) SIFT 1x1; global: (11) Color Layout, (12) Color Structure, (13) Dominant Color, and (14) Edge Histogram.

II. Instance Search

For instance search, we implemented two different subsystems. One uses bag-of-features of Color-SIFT, extracts and indexes the descriptors from the database in advance, and is thus able to provide results with very short query times. The other does not perform any preprocessing, but extracts and matches SIFT descriptors extracted from DoG points at query time.

A. Indexed Color-SIFT

BoF Features are generated in the same way as described for SIN in Section I-A2 using densely sampled regions from three different scales and Difference-of-Gaussian (DoG) interest points [13], Color-SIFT features, and codebooks with 1000 clusters that are generated using the k-means algorithm. The main difference between the dense sampled features and the features from DoG points is the fact that the latter ones are extracted in an orientation invariant way according to the dominant orientation of the DoG points. On the other hand, approximately the same number of Color-SIFT features are extracted in both approaches as a high-contrast filter limits the number of DoG points to a maximum of 300. In an offline process, global BoFs are then generated for both feature types from every clustered key frame. BoF matching is then performed with k-NN search of the BoFs from each query image against the BoFs from the clustered key frames. Matching is performed with histogram intersection between a query BoF Q and a clustered key frame BoF T as follows:

$$d = \sum_{i=0}^{i<N_{cb}} \max\left(Q\left(i\right), T\left(i\right)\right) - T\left(i\right) \tag{3}$$

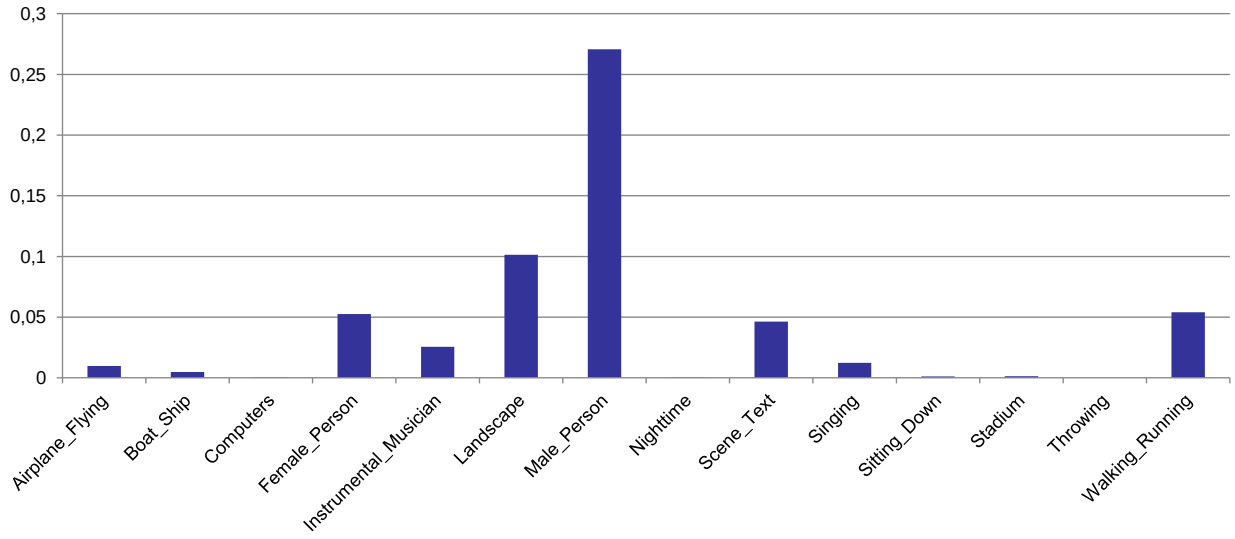

Fig. 2. Results of the SIN light run (infAP).

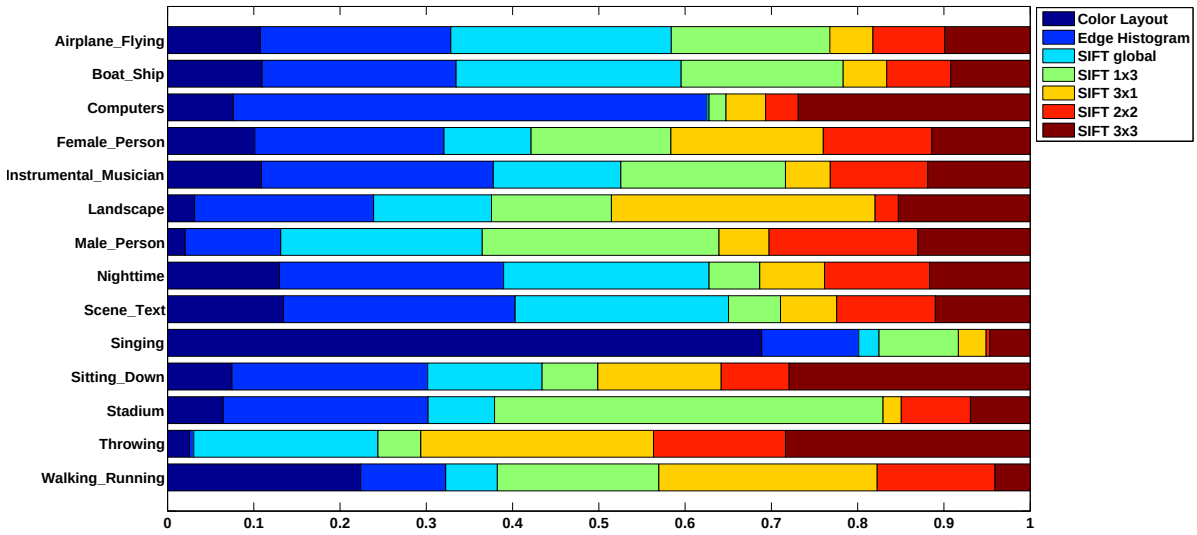

Fig. 3. Weights of the subkernels of the MKL problem by different features.

where N_{cb} is the size of the codebook and i is the current codebook index. This histogram intersection approach is used because the (cropped) query images contain only the query object while additional background objects can be shown in the clustered key frames.

B. SIFT matching at query time

SIFT [13] matching at query time is used as another subsystem in the instance search task. No preprocessing is done, but all feature extraction and matching is done at query time. For each query versus database image match, DoG keypoints and their corresponding SIFT descriptors are extracted as proposed in [13]. Only one field of the input image is used in order to avoid possible side effects of interlaced content. Descriptors are extracted from every frame of the video, as experiments on development data showed that short occurrences might be missed when using temporal subsampling, especially in case of camera motion, where motion blur or encoding artifacts might prevent some of the detections. The matching of the descriptors has been implemented on GPU using NVIDIA Cuda [2] in order to speed up processing. The minimum number of matching descriptors has been experimentally set to 8, and a confidence score is determined from the number of matching descriptors per frame and the number of frames which have been found to be matching.

[2]http://www.nvidia.com/object/cuda_home_new.html

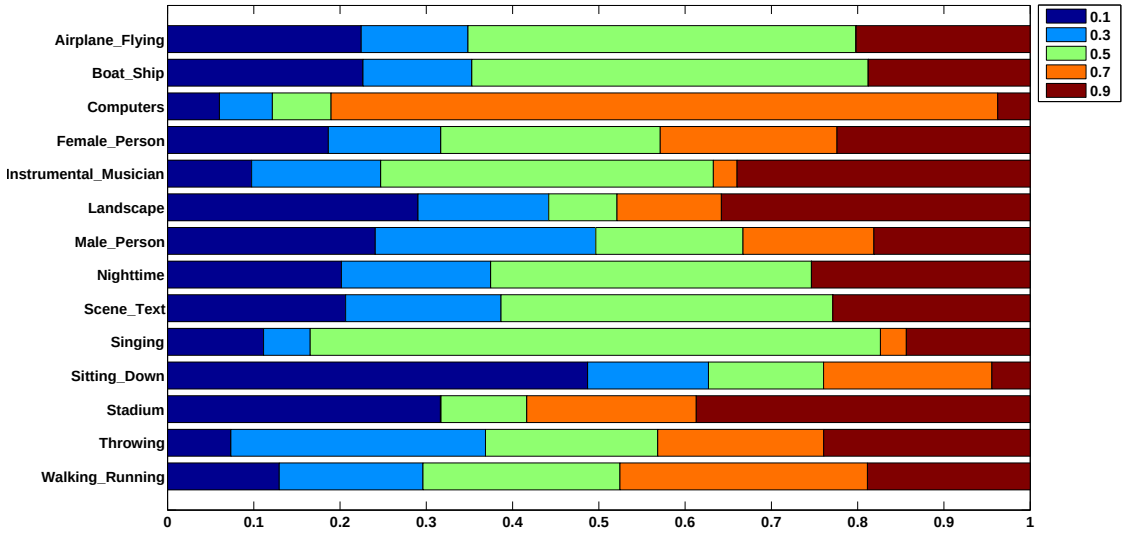

Fig. 4. Weights of the subkernels of the MKL problem by different values of the similarity threshold θ_{sim} of the LCSS kernel.

C. Fusion

Based on the observation that the SIFT matching approach tends to produce typically few results, with correct ones at the top of the list, the fusion method was designed as follows. From the results returned from SIFT matching, a threshold is estimated as the score with the steepest gradient at the lower third of the range of score values list. The results from SIFT matching with scores below this threshold are discarded, and the rest of the result list with the results from indexed Color-SIFT matching.

D. Results

We have submitted four runs:

- JRSVUT1 uses only the indexed Color-SIFT descriptors extracted from densely sampled points.
- JRSVUT2 uses only SIFT matching at query time.
- JRSVUT3 fuses the top results of SIFT matching at query time and the indexed Color-SIFT results extracted from densely sampled points.
- JRSVUT4 fuses the top results of SIFT matching at query time and the indexed Color-SIFT results extracted from DoG points.

The results of the four runs are visualized in Figure 5. The best results for queries 9053, 9057 and 9058 are at or close to the best result. The fast approach with indexed Color-SIFT turns out to be not discriminative enough for this type of queries, performing worse than a very similar approach used for INS in 2011. The SIFT matching at query time performs quite well overall, providing few but mostly correct results for many queries. Figure 6 shows the number of hits at rank 10, 30, 100 and 1000 for run JRSVUT2. There are three groups of queries: (a) the number of hits increases with the number of results (saturating below 100 in most cases), (b) some hits at the top of the list but no more are found, and (c) no correct hits at all.

For 9 of the queries, one or both of the fusion methods yield slightly better average precision by adding more hits below the top results. However, for five queries the fusion methods drops too many correct results from the SIFT matching results, thus decreasing the average precision. There are two different cases of this issue: For some queries the threshold chosen by the fusion method is too high, while for others the scores of the results are not sufficiently discriminative, containing a mixture of true and false positives around the threshold value. The mean average precision over all the queries is slightly lower for the two fused runs than that of run JRSVUT2.

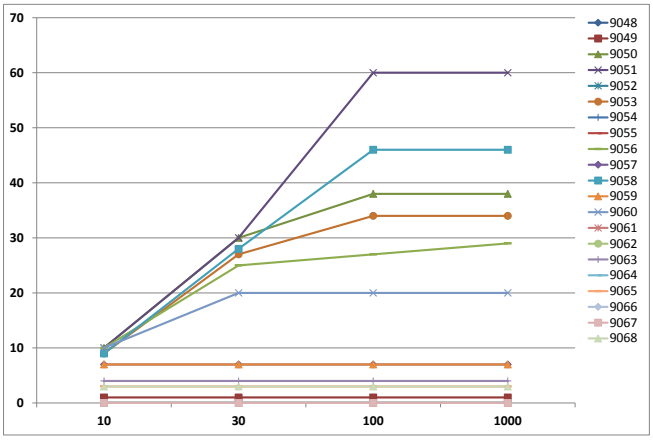

Fig. 6. Number of hits found for run JRSVUT2 at rank 10, 30, 100 and 1000.

When we analyze the queries with low performance, we see mainly two causes: A very low number of reliable interest

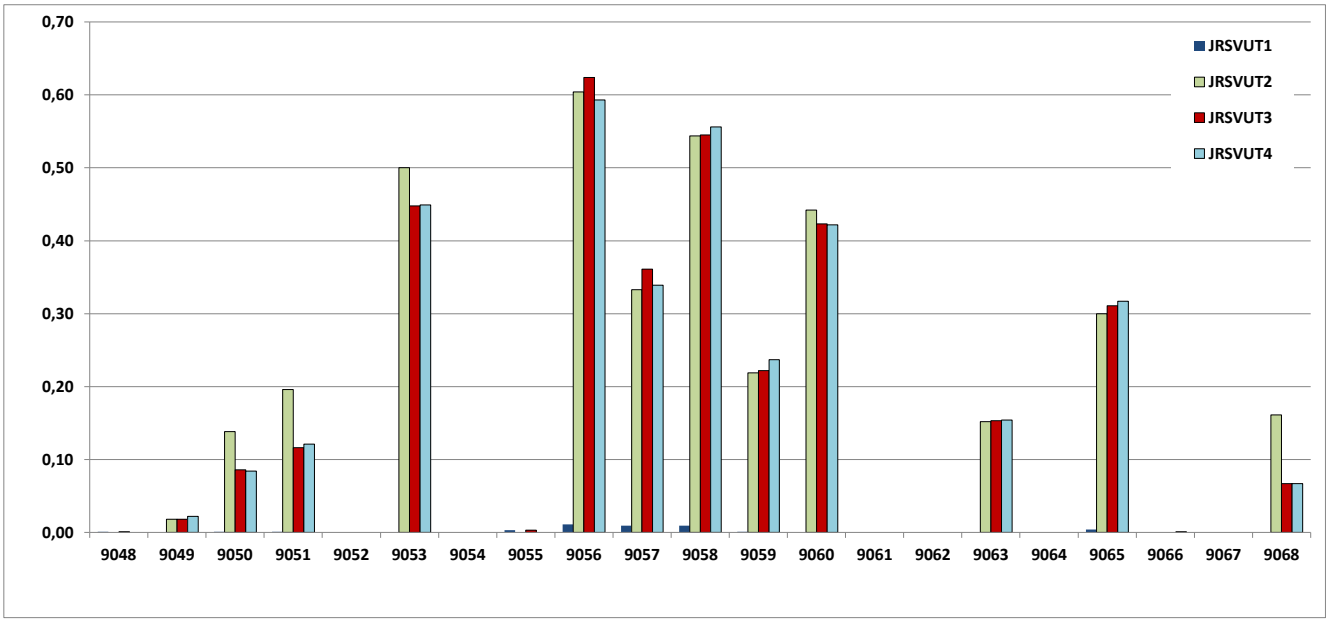

Fig. 5. Average precision of the four INS runs.

points or not sufficiently discriminative feature points. The first type of error is evident for some of the logos such as the Mercedes star (9048) and the London underground logo (9052), which have query samples with very low resolution. This is also true for the Pepsi logo (9061), which has few interest points even on samples with higher resolution. The second type of errors affects the Stonehenge (9054) and Hoover Dam (9066) queries, where most of the extracted feature descriptors match with any rock/concrete object. For the Empire State bulding (9064) and Sears Tower (9055) query we can observe both issues, the samples have rather low resolution, and the extracted descriptors match most similar buildings. The resolution of the samples was too low to obtain descriptors capturing the specific characteristics of this building. The MacDonald's logo (9067) is a special case, as the samples contain differently illuminated versions of the logo. The descriptors rather capture the lighted areas on dark background aspect than the shape of the logo, for example, the results contain many concert shots with different light effects on the stage.

E. Conclusion

The method using indexed Color-SIFT is very fast (query time well under 60 seconds for most of the queries), but the results are poor. The SIFT matching at query time provides good results. For some queries the results are at or close to the best results. Queries with small sample images or no distinctive visual properties yield very low performance. Fusion improves results for many of the queries, but removes a large number of correct hits for few of the queries.

ACKNOWLEDGMENTS

The authors would like to thank Christian Schober, Felix Lee, Georg Thallinger, Werner Haas and Horst Eidenberger for their feedback and support.

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant agreements n° 248138, "FascinatE – Format-Agnostic SCript-based INterAcTive Experience" (http://www.fascinate-project.eu/) and n° 287532, "TOSCA-MP - Task-oriented search and content annotation for media production" (http://www.tosca-mp.eu), as well as from the Austrian FIT-IT project "IV-ART – Intelligent Video Annotation and Retrieval Techniques".

REFERENCES

[1] "Information technology-multimedia content description interface: Part 3: Visual," ISO/IEC 15938-3, 2001.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[3] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1582–1596, 2010.

[4] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] W. Bailer, "A feature sequence kernel for video concept classification," in *Proceedings of 17th Multimedia Modeling Conference*, Taipei, TW, Jan. 2011.

[6] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, Dec. 2004.

[7] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra, "Video event classification using string kernels," *Multimedia Tools Appl.*, vol. 48, no. 1, pp. 69–87, 2010.

[8] P. V. Gehler and S. Nowozin, "Let the kernel figure it out; principled learning of pre-processing for kernel classifiers," in *CVPR*, 2009, pp. 2836–2843.

[9] D. Djordjevic and E. Izquierdo, "Relevance feedback for image retrieval in structured multi-feature spaces," in *MobiMedia '06: Proceedings of the 2nd international conference on Mobile multimedia communications*. New York, NY, USA: ACM, 2006, pp. 1–5.

[10] F. Odone, A. Barla, and A. Verri, "Building kernels from binary strings for image matching," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 169–180, 2005.
[11] S. Sonnenburg, G. Raetsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc, "The SHOGUN Machine Learning Toolbox," *Journal of Machine Learning Research*, vol. 11, pp. 1799–1802, June 2010.
[12] S. Sonnenburg, G. Raetsch, C. Schaefer, and B. Schoelkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, July 2006.
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.