Object instance recognition using motion cues and instance specific appearance models

Arne Schumann

Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany

arne.schumann@iosb.fraunhofer.de

ABSTRACT

In this paper we present an object instance retrieval approach. The baseline approach consists of a pool of image features which are computed on the bounding boxes of a query object track and compared to a database of tracks in order to find additional appearances of the same object instance. We improve over this simple baseline approach in multiple ways: 1) we include motion cues to achieve improved robustness to viewpoint and rotation changes, 2) we include operator feedback to iteratively re-rank the resulting retrieval lists and 3) we use operator feedback and location constraints to train classifiers and learn an instance specific appearance model. We use these classifiers to to further improve the retrieval results. The approach is evaluated on two popular public datasets for two different applications. We evaluate person re-identification on the CAVIAR shopping mall surveillance dataset and vehicle instance recognition on the VIVID aerial dataset and achieve significant improvements over our baseline results.

Keywords: object instance recognition, re-identification, retrieval, persons, vehicles, appearance model

1. INTRODUCTION

Object instance recognition is the task of finding additional ocurrences of a specific query object instance in image or video data. The most prominent example for this task is that of person re-identification. With constantly growing amounts of image and video data, the way this data can be searched and is presented becomes increasingly more important. Manual searches in such data by a human user or operator are impractical, because they take a long time, cause high personnel costs and are prone to errors due to the limited attention span of humans. Consequently, in recent years there has been a lot of research with the goal of providing automatic approaches to help solve this problem. Object instance recognition is a complex and challenging task. Approaches have to deal with differences in illumination, differences in cameras, different viewpoints, occlusion and poses (in case of non-rigid objects). The number of possible applications is large and ranges from person/actor retrieval in multimedia data over identification of suspects in surveillance data to retrieval of objects of interest in aerial video data.

Due to the high complexity of the task, we employ a semi-automatic approach that iteratively incorporates user or operator feedback to improve results. Initially, a first result list of object instances is automatically retrieved from a database by matching a set of image features. This list is then presented to an operator for manual annotation of some of the entries. Based on these annotations, we train instance specific appearance classifiers and re-rank the list to improve accuracy. This process can be iteratively repeated multiple times. An overview of the steps in our approach is depicted in Figure 1. We evaluate our approach for the scenarios of person retrieval in a surveillance setting and vehicle retrieval in aerial video data.

The remainder of this work is structured as follows. We present a brief overview of related work in Section 2. Sections 3, 4 and 5 describe our baseline approach, the use of operator feedback and instance specific object classifiers, respectively. We evaluate our approach on two datasets in Section 6 and conclude in Section 7.

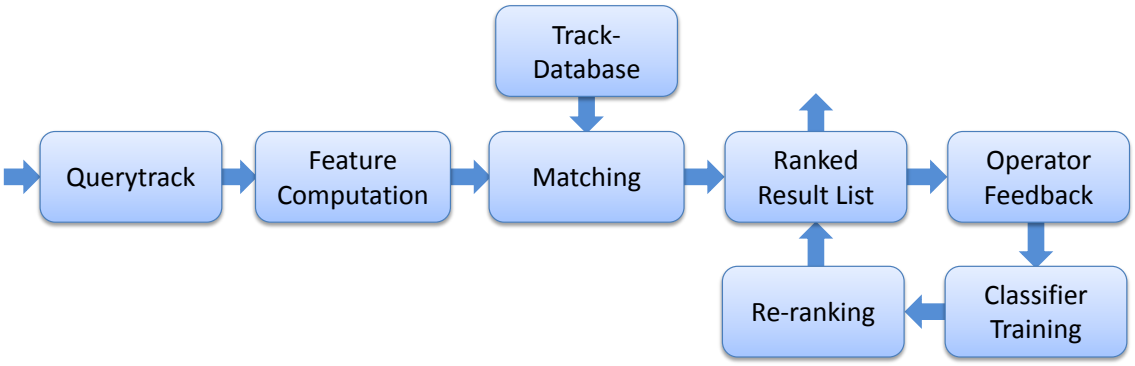

Figure 1: Overview of the object retrieval approach. Features are computed on a query track, matched to tracks in a database and sorted into a result list. The list can be iteratively improved by incorporating user feedback and object instance classifiers.

2. RELATED WORK

In Section 6 we evaluate our approach for person re-identification in surveillance data and vehicle recognition. Our discussion of related work is thus focused on those two areas.

There has been extensive work on person re-identification in recent years. We limit our discussion here to those approaches that are most closely related to our work.
Farenzena et al.[1] use a combination of color and texture features for appearance based person re-identification. They compute features on different body parts and exploit symmetry and asymmetry perceptual princples of the human body. Their approach can handle low resolution data, variation in viewpoints and illumination changes. Gray et al.[2] use an ensemble of localized features to achieve viewpoint invariance for pedestrian recognition. They design a number of color (RGB, HSV) and texture (Schmid, Gabor) feature channels and apply a boosting approach to select channels and locations for a discriminative appearance model. Wang et al.[3] use a part based approach to segment the image of a person and model the spatial distribution of the appearance relative to each of the object parts using HOG features in Log-RGB color space and coocurrence matrices. Prosser et al.[4] transform the re-identification problem into a ranking problem and learn a subspace in which the correct match should correspond to the highest rank. The ranking problem is then solved using an Ensemble RankSVM. Hirzer et al.[5] rank person images using a set of covariance feature descriptors. In a second stage, they train a discriminative classifier using boosting to obtain a refined ranking.
There has also been a number of works that use constraints to improve re-identification results. Bäuml et al.[6] use contextual constraints between tracks to improve feature matching scores through metric learning. They include operator feedback and use it to improve their approach in multiple stages. Yu et al.[7] use constraints to improve tracking of people in a smart-room environment via spatial clustering.
A more detailed discussion of appearance-based person re-identification can be found in Doretto et al.[8]

The amount of realted work for the task of vehicle recognition is more moderate. Many approaches rely on vehicle plate reading. These will not be discussed here, because they are very object specific and rarely practical in aerial video data. Vehicle recognition comes very close to being a classification task. Classification of make, model and color identifies most vehicles as accurately as possible. The corresponding classification literature will also not be discussed here, because this approach is again quite object specific and not related to our work.
However, there are some works that match our vehicle recognition scenario more closely. Shan et al.[9] match vehicles between two cameras by representing each vehicle ocurrence as an embedding of vehicle images from that same camera. The embeddings are then matched by a same-vs-different classifier to avoid an error prone direct matching of two vehicles. Similarly, Guo et al.[10] use vehicle prototypes for an embedding approach.

3. INSTANCE RECOGNITION

In order to compare two object tracks t_i and t_j during the retrieval task, we rely on a pool of image features. The features were chosen to capture different aspects of an object's appearance.

3.1 Color Features

Color information is usually one of the most discriminative cues to differentiate between two different instances of the same object class.

Color Histograms (CH) We use color histograms to encode the color distribution of an object. The histograms are computed on various color channels (RGB, HSV and Lab) using $8 \times 8 \times 8$ bins. While histograms can be computed quickly, they only contain information about the relative amount of each color on the object's surface and none about the spatial distribution or structure of colors.

Color Structure Descriptor (CSD) In order to provide this complementary information, we also compute color structure descriptors.[11,12] This descriptor is also a histogram and computed by moving a sliding window of size 8×8 over the object bounding box. At each window position the bins which correspond to a color inside the window are increased by 1. The use of the sliding window leads to a degree of spatial information in the resulting histogram. We compute the CSD feature on a slightly modified version of the HMMD color space, following the approach described in Hähnel et al.[11] We use 60 bins in the CSD histogram.

3.2 Texture Features

The second set of features we compute is meant to capture information of an object's texture.

Gabor Filter Responses (GFR) We compute responses for a bank of 40 gabor filters for each object. The filter bank consists of 8 different orientations and 5 scale levels. The 40 filter responses for each object are aggregated using either the maximumvalue to form a 40-dimensional descriptor.

Local Binary Patterns (LBP) Local binary patterns have been shown to work well for texture classification.[13] We use histograms of local binary patterns. Each bin in a histogram corresponds to one of the 36 rotation invariant values for $LBP^{ri}_{8,R}$. Histograms are computed on three different scales $R = 1, 2, 3$ and concatenated to a 108-dimensional descriptor.

3.3 Motion Cues

Object motion direction in the scene can be leveraged to increase the matching performance of color and texture features.

In the surveillance scenario we use the motion direction of a track to determine the orientation of a person relative to the camera. We differentiate only between frontal, backside, profile or uncertain orientation and assign a label for the closest matching orientation to each image of each track. When a person stands still, it is assigned the last known orientation. Tracks that contain no movement are assigned an uncertain orientation label. Profile orientations are flipped so that the person always faces left. Most tracks in our data only contain one dominant orientation. When comparing two tracks using the color and texture features, the orientation information is used to weight the matching scrores between images of the two tracks. The feature difference between images which have the same orientation is weighted more strongly than that of images which have a different orientation. This requires person images with the same orientation to match more closely while person images of different orientation need not match as exactly to still receive a good (low) matching difference. This approach requires the reasonable assumption that persons walk the way they face (i.e. not backwards).

Motion information is even more important in aerial video data. This type of data does not cause any problems with varying view angles (objects are always seen from above). However, depending on the motion directions objects in aerial video can have different rotations. This is especially problematic for those features that do not offer any inbuilt rotation invariance, such as CSD and GFR. In order to alleviate these problems, we rotate objects according to their motion direction before computing any image features. Both uses of motion information are depicted in Figure 2.

Figure 2: Motion information is used to focus on matching person images of the same orientation and to normalize rotation in aerial images of vehicles.

3.4 Location Constraints

Aside from motion information, we can also use the location of objects to increate matching performance. Objects which appear in the same image cannot belong to the same object instance. Thus, we forbid these objects to match. These location constraints will also be used to help train instance specific appearance classifiers (see Section 5).

In order to compare two tracks, we first normalize all computed feature vectors to unit length. We then compute the euclidean distance for each feature and between each image of track t_i and each image of track t_j. In the case of person surveillance data the distances are then weighted according to the difference in orientation. Finally, we choose only the minimum distance between the two tracks for each feature and use a linear combination of the feature distances to obtain a track distance:

$$d(t_i, t_j) = \sum_{k=0}^{3} \alpha_k \min_{l=1..|t_i|} \min_{m=1..|t_j|} \sqrt{f_k^{l\top} f_k^m} \tag{1}$$

where k is the index for our four features, α_k is the feature specific weight, $|t_i|$ denotes the number of images in track t_i and f_k^l corresponds to the feature with index k computed on the lth image of track t_i. The tracks in the database are then sorted according to their distance to the query track and presented to the operator.

4. OPERATOR FEEDBACK

Once a first result list has been automatically generated by the approach described in Section 3, we switch to a semi-automatic approach to further improve the results. The list is presented to an opertator who can label entries as correct or false. The list is then re-ranked based on this feedback. This procedure can be applied iteratively until a required accuracy is reached or the results no longer improve. A diagram of this approach can be seen in Figure 1.

In order to re-rank the list based on operator feedback, we re-weight the influence different features have on the track distances in Equation 1. Over all tracks that were newly annotated by the operator, we compare the distance values of the different features. Features correspond to the operator's feedback, if they lead to high distances for tracks which the operator marked as wrong or low distances for tracks which the operator marked as correct. For features f_i that correspond to the operator feedback, we raise the weight α_i. Features that do not correspond to the feedback are penalized by lowering their weight and thus decreasing their impact on the list ranking. This re-weighting adapts the track distance computation to the operator's observations.

Figure 3: Impressions from the datasets used to evaluate our approach. CAVIAR surveillance data on the left and VIVID aerial data on the right.

We assume the operator feedback to be free from errors. Thus, we can use it in combination with the location constraints to eliminate those tracks that were in the same image as one which was labelled correct by the operator. Tracks that were labelled as incorrect are eliminated as well.

5. INSTANCE APPEARANCE CLASSIFIERS

The approach described in the previous two sections already adapts itself to the specific object instance in the query track by adjusting feature weights using operator feedback. However, the weights can only affect entire features. Using a classifier trained for the specific object instance which is queried, we can learn which parts of a feature are helpful for the retrieval task and which are not. The disadvantage in using classifiers is that they require training data which is often not available. In this scenario, however, we already have two sources of such training data. We can use the query track as a positive example and all tracks which appear in the same images as negative samples. Additionally, we can use the tracks annotated by the operator as further positive and negative samples. And finally we can combine the operator annotations and the location constraints to obtain further negative samples.

Using this training data, we can train a classifier that differentiates between the query object instance and other object instances. We use a RBF-kernel Support Vector Machine (SVM) as our instance classifier and retrain after each iteration of operator feedback. The SVM is then run on the re-ranked result list. The classification results are substracted from the track distances, because negative classification scores indicate an object instance which is not the query instance and thus the track distance should grow larger:

$$d(t_{query}, t_i) = -\beta s_{query}(t_i) + \sum_{k=0}^{3} \alpha_k \min_{l=1..|t_{query}|} \min_{m=1..|t_i|} \sqrt{f_k^{l\top} f_k^m} \tag{2}$$

where $s_{query}(t_i)$ is the score of the query instance classifier on track t_i and β is a weight for that score.

6. EXPERIMENTS

6.1 Datasets

We use two publicly available datasets for two different applications to evaluate our approach. Figure 3 gives an impression of the different types of data.

CAVIAR Dataset The CAVIAR dataset* consists of several surveillance video sequences recorded by static cameras in a corridor of a shopping mall. The video data has a very low resolution of 384×288 pixels. We use ground truth annotations to generate tracks in order to avoid any negative impacts of tracking errors. Since there are many persons that apprear only once in all the 26 video sequences, we cut long tracks into pieces to artificially generate more tracks and ensure that each person appears in at least three tracks. When splitting a ground truth track into parts, we make sure to leave gaps of at least 100 frames between the parts. We end up with more than 200 tracks for a total of 64 individuals across all sequences.

VIVID Dataset The VIVID dataset[†,14] contains vehicles recorded from an aerial perspective. The dataset was originally recorded to evaluate tracking approaches and features color and infrared sequences. We focus only on the color video sequences. We extended the annotations from one vehicle per sequence to all appearing vehicles and again split longer tracks to increase the number of tracks per vehicle. The resulting dataset contains 15 different cars from multiple view points, various backgrounds and illumination changes.

6.2 Metric

We use the mean average precision (MAP) metric to evaluate our results. This metric is very popular for evaluating ranked retrieval lists. To compute the MAP we run our approach with each track in the dataset as a query track and generate a result list. For these result lists, we compute the average precisions (AP) and take the mean to get the MAP. The average precision of a list is computed as follows:

$$AP(L) = \frac{1}{\sum_{i=1}^{|L|} m_i} \sum_{i=1}^{|L|} m_i \frac{\sum_{j=1}^{i} m_j}{i} \tag{3}$$

where L is a ranked list of $|L|$ entries, m_i is 1 if the list entry at position i is a correct result and 0 otherwise. The first term of the metric normalizes the precision by dividing by the total number of possible correct tracks. The last fraction corresponds to the combined accuracy of the list entries from the top to position i. Thus, the higher a correct document is in the list L, the larger is its contribution to the performance metric.

6.3 Results

For our experiments we clip tracks by removing the first and last 10% of images in order to avoid situations where newly appearing objects are half cut off by the image boundaries. We shrink track bounding boxes in order to compute features only on the object itself and not on the background. For person tracks in the CAVIAR dataset, we chose the relative bounding box factors of $[x, z, w, h] = [0.3, 0.2, 0.4, 0.3]$ which focus on the upper body apprearance of a person. The bounding boxes are then normalized to a size of 64×128 pixels. The vehicle bounding boxes are chosen as $[x, y, w, h] = [0.2, 0.2, 0.6, 0.6]$ and normalized to a size of 128×64 pixels. We empirically determined that color histograms work best when applied to the HSV color space. The classifier weight of β is set to 0.1 for all experiments in order to avoid a too strong impact of the classification score in list ranking. We simulate operator feedback by using object id ground truth to label the first ten unlabelled results of a list in each iteration.

When using only a single feature for retrieval and the fully automatic approach without operator feedback and instance classifiers, the color structure descriptor turns out to be the most discriminative feature for both datasets. Results can be seen in Table 1. LBP histograms perform slightly worse and are almost on par with color histograms while gabor filters achieve the lowest performance. An equally weighted combination of all features results in a significant boost over any of the individual feature's performances.

Adding motion cues further increases the automatic retrieval performance for person retrieval on the CAVIAR dataset. However, the difference is even more significant on the VIVID dataset, because adding rotation invariance to CSD and GFR leads to a boost in MAP from 0.301 to 0.331. The results are shown in Table 2. Location constraints do not lead to any significant improvement for the automatic system. Their main benefit is providing data for instance classifiers.

*http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1

†http://vision.cse.psu.edu/data/vividEval/

Feature	CAVIAR	VIVID
CH	0.243	0.242
CSD	0.264	0.278
GFR	0.191	0.203
LBP	0.240	0.231
CH+CSD+GFR+LBP	0.295	0.304

Table 1: Mean average precision on both datasets unsing individual features. A combination of features increases performance significantly.

	CAVIAR	VIVID
No motion cues	0.295	0.304
Motion cues	0.301	0.331

Table 2: The use of motion cues further improves MAP. Especially the added rotation invariance in aerial video data if beneficial.

	CAVIAR				VIVID			
	automatic	1xFB	2xFB	3xFB	automatic	1xFB	2xFB	3xFB
CH+CSD+GFR+LBP	0.301	0.379	0.441	0.493	0.331	0.425	0.496	0.523
Classifier	0.301	0.389	0.455	0.508	0.331	0.431	0.501	0.532

Table 3: MAP results over multiple rounds of operator feedback (FB). Adding instance specific appearance classifiers results in the best performance.

Using the automatic approach as a reference, we evaluated performance over multiple rounds of user feedback. Results can be seen in Table 3. Re-weighting features though operator feedback can improve person retrieval performance on the CAVIAR dataset from an initial MAP of 0.301 to a MAP of 0.493 after three rounds of feedback. A similar impact can be observed on the VIVID dataset. Here, the increase in MAP is even stronger. This is likely due to the smaller amount of object instances in this dataset. The full approach using instance classifiers achieves consistently higher MAP values than using only the combination of features. Our best MAP results using the full approach are 0.508 and 0.532 on CAVIAR and VIVID, respectively, which is a significant improvement over the baseline automatic approach. Some results of our approach can be seen in Figure 4.

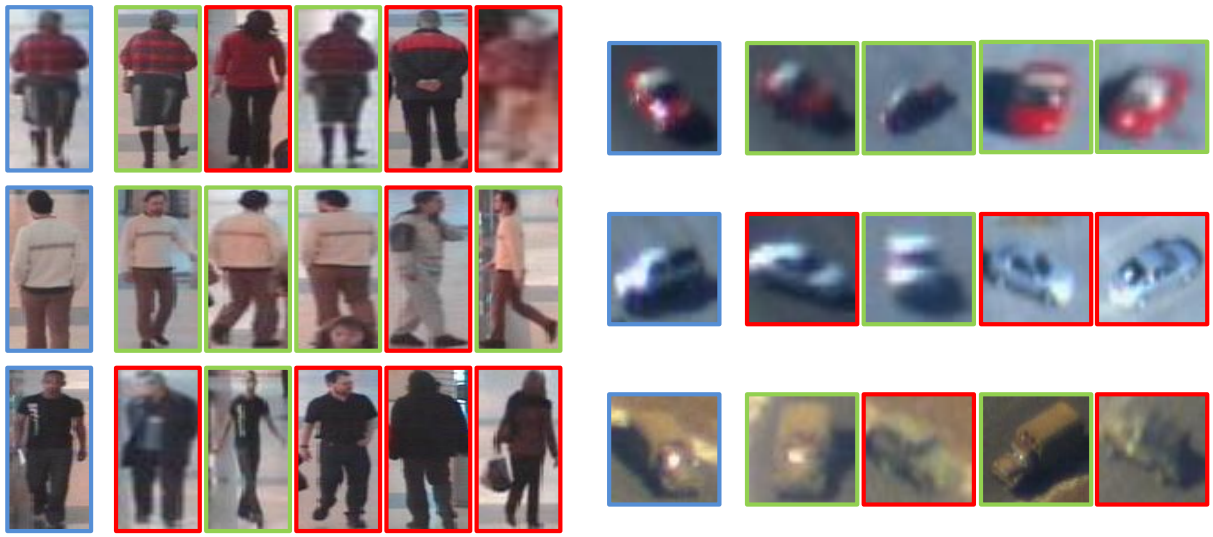

Figure 4: Top retrieval results for a sample set of query images (blue). Correct results are marked green, false results red.

7. CONCLUSION

In conclusion, we presented an automatic object instance retrieval approach using feature matching. We improved over the baseline automatic approach by using motion features to weight matching scores of person tracks in surveillance data and add rotation inviariance to features in aerial video data. By adding iterative rounds of operator feedback and training instance specific object classifiers, we were able to significantly outperform our initial restults. We achieve consistent performance on very different data and object classses, validating the versatility of our approach.

In the future we plan to further explore the topic by investigating more powerful machine learning techniques for the instance classifiers as well as additional ways to make use of operator feedback.

REFERENCES

[1] Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M., "Person re-identification by symmetry-driven accumulation of local features," in [*CVPR*], (2010).

[2] Gray, D. and Tao, H., "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in [*ECCV*], (2008).

[3] Wang, X., Doretto, G., Sebastian, T., Rittscher, J., and Tu, P., "Shape and appearance context modeling," in [*ICCV*], (2007).

[4] Prosser, B., Zheng, W.-S., Gong, S., Xiang, T., and Mary, Q., "Person re-identification by support vector ranking," in [*BMVC*], (2010).

[5] Hirzer, M., Beleznai, C., Roth, P. M., and Bischof, H., "Person re-identification by descriptive and discriminative classification," in [*SCIA*], (2011).

[6] Bauml, M., Tapaswi, M., Schumann, A., and Stiefelhagen, R., "Contextual constraints for person retrieval in camera networks," in [*AVSS*], (2012).

[7] Yu, T., Yao, Y., Gao, D., and Tu, P., "Learning to recognize people in a smart environment," in [*AVSS*], (2011).

[8] Doretto, G., Sebastian, T., Tu, P., and Rittscher, J., "Appearance-based person reidentification in camera networks: problem overview and current approaches," *AIHC* (2011).

[9] Shan, Y., Sawhney, H. S., and Kumar, R., "Vehicle identification between non-overlapping cameras without direct feature matching," in [*ICCV*], (2005).

[10] Guo, Y., Shan, Y., Sawhney, H., and Kumar, R., "Peet: Prototype embedding and embedding transition for matching vehicles over disparate viewpoints," in [*CVPR*], (2007).

[11] Hahnel, M., Klunder, D., and Kraiss, K.-F., "Color and texture features for person recognition," in [*IJCNN*], (2004).

[12] Ohm, J. R., Cieplinski, L., Kim, H. J., Krishnamacha, S., Manjunath, B. S., Messing, D. S., and Yamada, A., "Color descriptors," in [*Introduction to MPEG-7*], John Wiley & Sons, Inc. (2002).

[13] Ojala, T., Pietikainen, M., and Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *PAMI* (2002).

[14] Collins, R., Zhou, X., and Teh, S. K., "An open source tracking testbed and evaluation web site," in [*PETS*], (2005).