Can Motion Segmentation Improve Patch-based Object Recognition?

Adrian Ulges
German Research Center for Artificial Intelligence (DFKI)
adrian.ulges@dfki.de

Thomas M. Breuel
IUPR Research Group
University of Kaiserslautern
tmb@cs.uni-kl.de

Abstract

Patch-based methods, which constitute the state of the art in object recognition, are often applied to video data, where motion information provides a valuable clue for separating objects of interest from the background. We show that such motion-based segmentation improves the robustness of patch-based recognition with respect to clutter. Our approach – which employs segmentation information to rule out incorrect correspondences between training and test views – is demonstrated empirically to distinctly outperform baselines operating on unsegmented images. Relative improvements reach 50% *for the recognition of specific objects, and* 33% *for object category retrieval.*

1 Introduction

Over the last years, visual recognition methods have experienced a break-through due to novel strategies for the covariant detection and robust description of distinctive image regions [9, 13]. The resulting *patch-based* methods have been subject to extensive research, offering a high robustness with respect to clutter, variations of scale and perspective, as well as illumination changes [4, 8, 15].

In this paper, we study whether the robustness of patch-based object recognition can be increased further by a segmentation of objects of interest. While this turns out to be a difficult challenge for static images, motion information in video provides a strong clue for meaningful segmentations of real-world objects [3, 14].

On the one hand, such a segmentation could help rule out false positive correspondences caused by clutter. On the other hand, motion segmentation remains a challenging problem with potentially error-prone and inaccurate results, and background information might provide a valuable clue for the presence of an object (for example, visible roads hint at the presence of cars). Therefore, the influence of motion segmentation on patch-based object recognition needs to be studied.

To do so, we present a patch-based approach that uses motion segmentation as a filter for correspondences (Section 3). Using this framework, the combination of motion segmentation with object recognition is studied in two quantitative experiments (Sections 4 and 5) – one regarding the recognition of specific objects, the other object category retrieval in video databases. Our results demonstrate that motion segmentation – even if inaccurate – gives significant improvements over baselines operating on unsegmented images.

2 Related Work

While motion segmentation [3, 16] and object recognition [10, 11] have both been subject to extensive study, less work can be found on their combination.

Some approaches enhance segmentation with top-down object knowledge [2, 6], mostly by penalizing deviations from a pre-learned object shape. Such approaches also have also employed motion information in video [7], but are focused on improving segmentation, and their recognition capacity is usually not evaluated against competitive unsegmented baselines.

Kühne et al. [5] have fed foreground regions resulting from a motion segmentation to a shape-based recognition (again, no comparison with baselines free of segmentation was made). Rothganger et al. [12] have used structure-from-motion to infer 3D patch models, and demonstrated improvements over a patch-based recognition from 2D images. We validate similar benefits without a full 3D reconstruction, using a simpler and more efficient 2D motion segmentation.

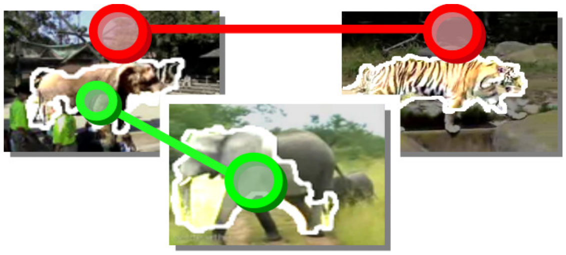


Figure 1. Our approach accepts local matches in object regions obtained from motion segmentation (green) but discards them in the background (red).

3 Approach

We follow a state-of-the-art patch-based approach in a sense that recognition is carried out as a search for correspondences between local features in the image to be recognized and in labeled training views [4, 8, 15]. Our key contribution is a filtering of the resulting correspondences on the basis of motion segmentation (see Figure 1 for an illustration).

Features and Matching As a robust patch-based feature representation, SURF features [1] are chosen (other detectors and descriptors [9, 13] could be used equivalently). We test two strategies for determining correspondences between patches x in a test image and patches x' in training images:

1) Full Patch Search: for each test patch x, the nearest neighbor x' among all training patches is found, and both are assumed to form a match.

2) Visual Words: a popular speed-up strategy is to cluster patches previous to recognition (typically using K-Means), obtaining *visual words* [15]. Two patches are then assumed to match whenever mapped to the same closest visual word.

Refinement by Motion Segmentation The correspondences we obtain are error-prone, containing false positives due to local similarities between different objects or clutter. To discard such erroneous matches, the core concern of this paper is to employ motion segmentation. Our approach first uses motion segmentation to segment dynamic scenes into a foreground and background layer. Inspired by Schöenemann's and Cremer's approach [14], both layers are assumed to move according to a parametric (affine or constant) motion model, and segmentation is carried out by an iterative refinement using graph cut. We further add a color extension such that pixels are mapped to layers not only based on their motion but also on their color [17].

This information is then used to filter correspondences: only matches are accepted for which both the training patch and the test patch belong to the foreground (i.e., object) region.

Other Refinement Strategies A variety of other refinement strategies have been suggested previously based on patch appearance and spatial arrangement. A key question is whether motion segmentation has the potential to substitute and complement these approaches. We compare our motion segmentation filtering with two other strategies:

1) Nearest Neighbor Ratio: This strategy rejects inconfident correspondences [8]: besides each nearest neighbor x', we also find a second nearest neighbor x'' from all patches with a different object label than x'. The match $x \sim x'$ is accepted only if the ratio $\frac{|x-x'|}{|x-x''|}$ is lower than a certain threshold $\lambda \in [0, 1]$ (i.e., if the match for one object is significantly closer than for all others). If $\lambda = 1$, no filtering takes place, and the further λ is decreased the more matches are rejected.

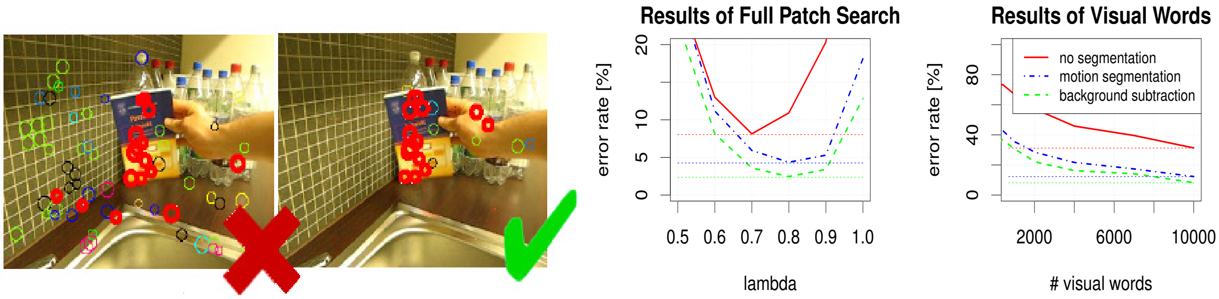
2) Spatial Constellation: While correct matches are spatially constrained, false positives tend to be scattered all over an image. To make use of this fact, features in the test view are assumed to map to their correspondences in training views by an affine transformation, which is estimated using an approach by Lowe [8]. Matches that deviate more than five pixels from the estimated transformation are discarded.

Scoring Finally, we use the (potentially refined) set of correspondences to infer object *scores* [4]. Matches $x \sim x'$ induce *votes* for the object labels of x', which are aggregated using a simple sum rule fusion: an object's score corresponds to the number of correspondences with its label divided by the overall number of correspondences. This score can either be used for recognition or for retrieval (i.e., to rank test items).

4 Experiment 1: Controlled Setup

The goal of the first experiment is to recognize objects presented to a camera. By using the same backgrounds for all objects, we decorrelate clutter from the object class. Also, a static camera is used, which allows us to obtain a close-to-perfect segmentation using background subtraction [17]. In this sense, our first setup is *controlled*.

The dataset for this experiment was manually generated by presenting 12 books to a camera at 12 indoor locations, obtaining 144 short video clips (sample in Figure 2(a)). $3-4$ keyframes were used showing each object at each location. We obtain about 600 SURF features on average per frame.



(a) (b) (c)

Figure 2. Results of Experiment 1 indicate improvements by using motion segmentation: (a) sample result without (left) and with (right) motion segmentation – red patches indicate correct matches. (b) Quantitative results of full patch search. (c) Results using visual words.

Our approach was tested using one-shot learning: videos taken at one location were used as training samples, i.e. their patches were stored and matched against the one from all other test videos. By repeating this experiment with all 12 locations, $1,584$ object decisions were obtained, over which we measure the error rate. Three setups were tested: one not using segmentation at all, one with our motion-based approach (Section 3), and one using even more accurate background subtraction that, however, requires a static camera.

Figure 2(b) illustrates results of full patch search (error plotted against the nearest neighbor ratio λ). We see significant improvements by nearest neighbor ratio filtering, but also further ones by motion segmentation: compared with unsegmented images, motion segmentation gives a statistically significant (t-test, level 99%) reduction of error from 8.1% to 4.4%, an even more accurate background subtraction (green) gives 2.5%. A sample result is illustrated in Figure 2, for which standard patch matching, despite filtering nearest neighbor ratio, suffers from false positive matches (which are filtered by motion segmentation).

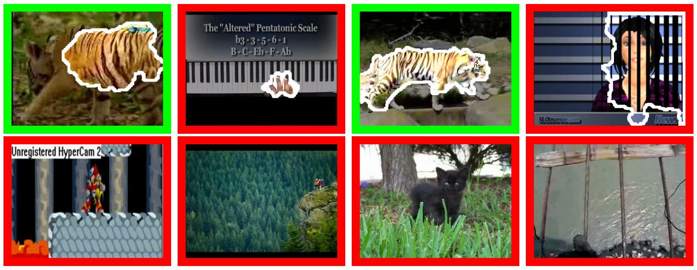
While nearest neighbor ratio and motion information resulted in a highly effective filtering, a refinement by spatial arrangement did not give any further improvements. Our impression was that – while in some cases this approach helped to filter outliers – it also tended to discard correct matches. For visual words (Figure 2(c)), which do not allow a filtering by nearest neighbor ratio, error is much higher. However, accuracy can again be improved by motion segmentation: for the best codebook size ($10,000$, which is comparable to using no discretization at all), a significant (t-test, level 99%) error reduction from 31.31% (no segmentation) to 12.31% (motion segmentation) is achieved.

5 Experiment 2: Object Retrieval

While Experiment 1 provided a *controlled* setup (with a static camera and decorrelated clutter), we also studied the proposed framework in a real-world scenario. As a test domain, the detection of four animal categories (“tiger”, “zebra”, “giraffe”, “elephant”) in video databases was chosen. Test content was downloaded from YouTube, small shots of $1-2$ seconds length were sampled, and the data was manually filtered such that only one shot per video was kept, showing a single moving animal at sufficient size. $29-50$ shots per animal category were obtained (160 shots total), and combined with $1,000$ random shots sampled from the YouTube-22Concepts Dataset showing none of the four animals[1]. Keyframes were sampled at regular steps of 6 frames, and votes for a video were accumulated from all keyframes. For efficiency reasons, only the visual words approach was tested (using $5,000$ clusters). Evaluation was done in a leave-one-out fashion: each shot was scored using the rest of the dataset as a training set. We measure retrieval performance by the average precision over the ranked retrieval list.

Results are illustrated in Figure 3: the top 4 detection results for the concept “tiger” are shown when using segmentation (top) and when not (bottom). While in the unsegmented case no hits are found, the system with segmentation detects two tigers correctly, and the other two detections show dark vertical bars similar to tiger fur. Quantitative results (Figure 3(b)) indicate that – for all animals – the average precision of retrieval can be improved significantly by segmenting foreground objects. Lower improvements for the “elephant” category can be explained by more frequent failures of seg-

[1] `http://madm.dfki.de/research/youtube-22concepts`



(a)

avg. precision (%)		
	segmented	
category	no	yes
tiger	15.6	**34.5**
zebra	27.2	**34.1**
giraffe	27.2	**42.5**
elephant	53.7	**55.3**
MAP	30.9	**41.6**

(b)

Figure 3. (a) The top 4 detection results for "tiger" with motion segmentation (top) and without (bottom). (b) Quantitative Results of Experiment 2 (pictures from YouTube).

mentation (often, only certain body parts like legs or trunks are segmented). Overall, mean average precision (MAP) is improved significantly (sign test over rank improvement, level 99%) from 30.9% to 41.6%. Random guessing on this challenging dataset would correspond to 3.5%. Note that results are improved despite a correlation between object category and clutter, which indicates that segmentation serves as a stronger clue for recognition than context (for manually segmented objects, Zhang et al. have made similar findings [18]).

6 Conclusions

In this paper, we have suggested motion segmentation as a filter for false correspondences in patch-based object recognition. Our empirical results demonstrate that even a simple approach (based on affine or translational motion) can improve robustness with respect to clutter, even in situations where clutter is correlated with the object class (see Experiment 2). Possible future directions along this line of research include benchmarks for more elaborate segmentation models (e.g., allowing the segmentation of multiple objects) and comparisons with sparse, feature-based approaches [16] and structure-from-motion representations [12][2].

References

[1] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded Up Robust Features. In *ECCV*, pages 404–417, 2006.

[2] E. Borenstein and S. Ullman. Class Specific, Top-Down Segmentation. In *ECCV*, pages 639–641, 2002.

[3] M. Irani and P. Anandan. About Direct Methods. In *IWVA*, pages 267–277, 2000.

[4] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search - Extended Version. Technical report, INRIA, RR 6709, 2008.

[5] G. Kühne, S. Richter, and M. Beier. Motion-based Segmentation and Contour-based Classification of Video Objects. In *ACM Multimedia*, pages 41–50, 2001.

[6] M. Kumar, P. Torr, and A. Zisserman. Extending Pictorial Structures for Object Recognition. In *BMVC*, pages 789–798, 2004.

[7] P. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, pages 18–25, 2005.

[8] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[9] K. Mikolajczyk. A Performance Evaluation of Local Descriptors. In *CVPR*, pages 257–263, 2003.

[10] J. Mundy. Object Recognition in the Geometric Era: A Retrospective. In *Toward Category-Level Object Recognition*, pages 3–28. Springer-Verlag New York, 2006.

[11] P. Roth. Survey of Appearance-based Methods for Object Recognition. Technical Report ICG-TR-01/08, Computer Graphics & Vision, TU Graz, 2008.

[12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects. *IEEE PAMI*, 29(3):477–491, 2007.

[13] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. *IJCV*, 37(2):151–172, 2000.

[14] T. Schönemann and D. Cremers. Near Real-Time Motion Segmentation using Graph Cuts. In *Proc. DAGM-Symposium*, pages 455–464, 2006.

[15] J. Sivic and A. Zisserman. Video Google: Efficient Visual Search of Videos. In *Toward Category-Level Object Recognition*. Springer-Verlag, 2006.

[16] P. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *IWVA*, 2000.

[17] A. Ulges and T. Breuel. Segmentation by Combining Optical Flow with a Color Model. In *ICPR*, 2008.

[18] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV*, 73(2):213–238, 2007.

[2] This work was supported by the German Federal Ministry of Education and Research, project INBEKI (13N10787), and by the German Research Foundation (DFG), project MOONVID (BR 2517/1-1).