Stable Interest Points for Improved Image Retrieval and Matching

Matthew Johnson and Roberto Cipolla
University of Cambridge

September 16, 2006

Abstract

Local interest points and descriptors have been used very successfully to achieve accurate and efficient image retrieval and matching performance which is robust to occlusion and limited viewpoint change. Currently, these systems tend to be initialized from still images and require that a thousand or more points be stored in a retrieval data structure for each object. Many of these points are rarely if ever used, and thus unnecessarily limit the number of reference images that can be stored effectively. We propose a method for determining the stability of local interest points and their descriptors such that an efficient and effective subset of points can be stored. This technique has been shown to reduce the number of required points by an order of magnitude while improving performance, allowing for significantly smaller data structures for use in retrieval and matching.

1 Introduction

There has been a good deal of work recently using interest points and robust, local descriptors for image retrieval and matching [12, 7, 11, 15, 6, 9, 13, 20]. The majority of these methods incorporate a nearest neighbor search, where the closest match for the descriptor at an interest point in the query image is found for the ultimate purpose of discovering which image or images in a database appear(s) in the query. Groups of these matches are verified, often using geometric constraints, and then evaluated; the final result is a ranked list of possible database images which appear in some form in the query. A sample result of first choices from such a system is shown in Figure 1.

Given that a nearest neighbor search is involved in almost all these techniques, there has been an accompanying burst in research on nearest neighbor data structures. While the nearest neighbor problem has been solved for quite some time in the low-dimensional case [4, 16, 17], the descriptors used in these systems tend to be quite large in dimension. The smallest are in the 30+ range, with many in the hundreds, thus suffering from the curse of dimensionality [2]. In this case, as the number of dimensions in the search space increases the cost of performing a nearest neighbor search using any of the clever low-dimensional structures approaches the cost of a linear search [5].

There have been many attempts to deal with this problem, mainly taking the form of improved data structures which perform an approximate nearest

Figure 1: *Sample Retrieval Result.* This is a sample query image, in which reference images from the database have been artificially modified and made to occlude each other to test retrieval in difficult situations. The rectangles around each image indicate that it has been correctly identified, with the smaller rectangles within each indicating what features were used in the matching process.

neighbor search, where the efficiency of the search is directly related to the uncertainty of the result. There has not been significant work, however, in reducing the number of points used. Our contributions in this paper are twofold:

1. A method for determining which interest points are most useful for storage, a condition we term "stability", in reference to the points being found in a relatively unaltered state in many different queries (§2).

2. A method for matching images and evaluating those matches in cases where there is a large proportion of incorrect descriptor matches that aids in determining stability (§3).

We show that it is possible to maintain and indeed improve performance with a significantly smaller number of well-chosen interest points and their descriptors, thus improving data storage efficiency and overall system performance for potentially all image matching and retrieval systems of this kind.

1.1 Previous Work

The use of local interest points for image retrieval was pioneered by Schmid and Mohr [18] and extended in many recent papers [7, 11, 15, 6, 9, 13, 20]. Several comparison studies of the various interest points and descriptors have been carried out by Mikolajczyk and Schmid, of which the most recent [14] is an excellent survey of the field. Of particular interest has been the SIFT system designed by David Lowe [12], which couples a descriptor that is relatively invariant under affine transformations and viewpoint changes with interest points found by localizing difference of Gaussian extrema in scale space. A similar system was extended by Ke and Sukthankar, called PCA-SIFT [8], which is of lower dimension (36 to SIFT's 128) but achieves similar performance.

All of these systems, when used in the context of image retrieval and matching, require a nearest neighbor search of some kind. In the case of matching two images, a linear search is often used. However, large scale systems must

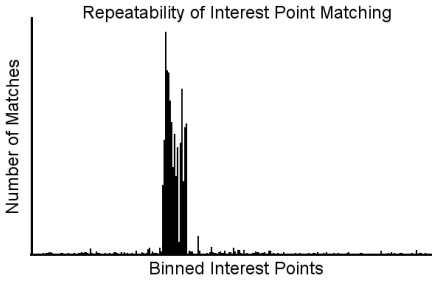

Figure 2: *Repeatability of Interest Point Matching.* In order to create this histogram, every point extracted from a still image of an object was matched against a 600 image training corpus for that object and every correct match (*i.e.* resulting in a correct pose prediction) recorded. There are 3656 points in 300 bins, and the points are in no particular order. Notice how there is a distinct fraction which is being repeatedly detected and matched, and a large number of points that are never seen in the training data.

employ more effective search structures. Lowe uses a KD-tree with a modified search algorithm he introduced with Beis in [1] called Best Bin First. Ke *et al.* introduced a system in [7] which uses a disk-based version of Indyk and Motwani's Locality Sensitive Hashing technique [5] for nearest neighbor search. Yet another approximate nearest neighbor data structure is the spill tree, a modified metric tree introduced by Liu *et al.* [11] which utilizes dimensionality reduction as well to achieve excellent results. All of these data structures show performance deterioration as the number of points stored in them increases. As such, a method of reducing the number of points needed for each reference image will increase the maximum number of images that can be stored while maintaining a desired level of quality.

No one has tried to cull the points which enter the data structure as we try to do. Lepetit *et al.* [10] warp patches to approximate affine changes in a still image and use the statistics of the various points to determine an efficient classifier, but it unclear how their system would scale to large numbers of images for retrieval databases. Corso and Hager attempt much the same thing as we do here in their impressive work in [3], though they use image regions obtained through segmentation to achieve the data reduction, whereas our technique is a proposed improvement to existing systems which use local interest points and requires no modification of those systems besides a culling step during database creation.

2 Stable Points

It is unnecessary to store every descriptor extracted from a still image of a desired object. They number in the thousands for a standard reference image (*i.e.* 640 pixels by 480 pixels) and many of them will be unused in matching. We have found that, on average, over half of these descriptors are never matched in queries and that of those which are, an even smaller subset account for the majority of matches as can been seen in Figure 2. It is the goal of this paper to lay out a process by which to determine which descriptors are in this small

subset and the best way to utilize this smaller subset in matching.

We begin by realizing that the best way to discover the properties of the descriptors extracted from a reference image of an object is to use them to match a set of query images and to experimentally determine which are used most often. This set of query images can take the form of a set of machine modified images (Ke *et al.* create a set of such images for their tests in [7]), a video taken of the object in its environment (*e.g.* of a painting in an art gallery) or a collection of sample queries for a database. The type of query image set used can be determined in an application-dependent manner, however each is an approximate sampling of the overall viewset of the object.

Once an appropriate set of query images has been collected, a nearest neighbor data structure is filled with the descriptors extracted from the reference image and each query image is matched using this database. After geometric verification, the remaining pairs are recorded. We used the KD-tree based system from [1] in this research, but any appropriate nearest neighbor structure will do, though a linear search can be time consuming for large query image sets. For verification, we used the system presented in §3.

With the matching pairs in hand, each descriptor from the reference image can now be ranked in terms of the number of queries in which it was found. In addition, all descriptors it was matched to are combined as a sample and the mean and diagonal covariance computed to describe the statistics of the descriptor as found in the queries. Once these have been found, the descriptors are assigned two real valued scores, one for repeatability and the other for deviation. The repeatability score is computed as

$$\text{Repeatability} = \frac{f_i - f_{min}}{f_{max} - f_{min}} + 1 \tag{1}$$

where f_i is the number of frames in which the descriptor i was found, $f_{min} = \text{argmin}_i(f_i)$ and $f_{max} = \text{argmax}_i(f_i)$. Deviation is computed as

$$\text{Deviation} = \frac{d_i - d_{min}}{d_{max} - d_{min}} + 1 \tag{2}$$

where $d_i = \sqrt{|V_i|}$. V_i is the covariance of the descriptor i, $d_{min} = \text{argmin}_i(d_i)$ and $d_{max} = \text{argmax}_i(d_i)$. The stability is thus computed as

$$\text{Stability} = \frac{\text{Repeatability}}{\text{Deviation}}. \tag{3}$$

The stability measure is designed to give a preference for descriptors which are extracted from interest points that are detected in many different views of the object and which do not exhibit much variance in description. These points should be given preference in storage, and indeed we show that by using subsets consisting of descriptors with high stability we can achieve improved performance to that of using the entire set of reference descriptors. It is reasonable to assume that the kinds of transformations used in the training data will create a preference for certain kinds of points over others, and thus they should ideally be representative of those transformations that would be present in the desired application domain.

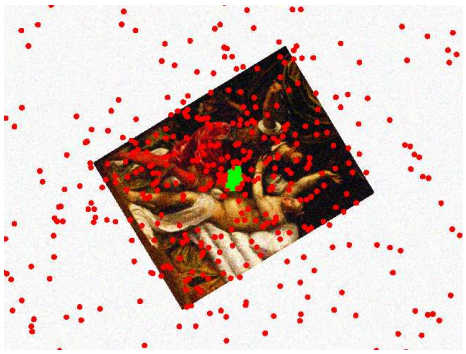

Figure 3: *Center Clusters.* Each circle shown is a proposed center from a keypoint based on that keypoint's scale, orientation, location and the relative position from the center of its matched keypoint in the database. As can clearly be seen, there is a clear cluster of correct points in the case of a true positive and just noise for false positives. The color and shape of the keypoints indicate which mixture component they belong to, again notice that the noise in the image is explained completely, with the correct matches being identified clearly for use in geometric verification.

3 Probabilistic Pose Prediction

In order to determine which matching pairs of points in a query image are valid, a pose for the object must be determined in the query. Only pairs agreeing with this pose will be retained and used in stability analysis. Instead of the generalized Hough transform used by Lowe, which produces binning effects that can result in incorrect matches and pose predictions, we describe here a novel probabilistic approach to give a candidate pose similar in concept to that used by Seeman *et al.* [19] for pedestrian detection.

For every point in a nearest neighbor database, we have encoded position as a vector pointing from the center of the database image to the feature, denoted x. For every matching pair, we calculate the difference in orientation, $\Delta\theta = \theta_q - \theta_r$ and the difference in scale $\Delta s = s_q/s_r$ and use them to create a transformation matrix T where

$$T = \begin{bmatrix} \Delta s\cos(\Delta\theta) & -\Delta s\sin(\Delta\theta) \\ \Delta s\sin(\Delta\theta) & \Delta s\cos(\Delta\theta) \end{bmatrix}. \tag{4}$$

With T, we can now find a vector $\mathrm{x}' = T\mathrm{x}$, with which we can find the predicted center of the source image in the query image by subtracting x' from the query point position. Thus, every match pair predicts a center, and we can analyze this plot to group the pairs. Correct poses will correspond to tight distributions of center predictions which are produced by the true keypoints generated by the object, while other predictions come from false matches and thus resemble uniform background noise, as can be seen in Figure 3. The most natural way to solve the problem is by fitting a mixture model of a bivariate Gaussian distribution and a uniform distribution to the plot. Those points which belong to the Gaussian mixture component can then be used to find an affine transformation between the two images.

Once a transformation has been obtained it must be evaluated for likelihood. We are interested in the joint probability of the transform and the data,

$p(A, D) = p(A|D)P(D)$, where A is the affine transformation of the pose and D is the set of matching points. To do this, we model the posterior $p(A|D)$ as a bivariate Gaussian, the mean and covariance of which are calculated using the center point predictions of the matching points. $p(A|D)$ is then calculated by using the affine matrix's center prediction, $\begin{bmatrix} t_x \\ t_y \end{bmatrix}$. For correct matches, the transform's center prediction is at or very near the mean and thus gives a high probability. The prior $P(D)$ is taken as the percentage of total matching points remaining after fitting represented by the basis points D. For incorrect matches, the transform's center prediction is far from the mean and given the distribution's large variance in these cases, results in a very low probability for the match, with the prior ensuring that poses with a small number of corresponding basis points will not be considered as likely as those with more support. The pose with the highest likelihood is chosen, and its basis recorded for stability purposes.

4 Results

We performed experiments with 5 different paintings in 5 different environmental settings. For each setting a video of the painting was recorded using a web camera in which the painting undergoes a series of affine transformations. The settings were as follows:

Setting	*Camera*	*Painting*	*Lighting*
1	1, stationary	in book, free motion	artificial
2	1, stationary	in book, free motion	natural
3	1, freehand motion	in book, stationary	artificial
4	2, freehand motion	in book, stationary	natural
5	3, freehand motion	in museum, stationary	interior

The first setting was used as training, with the others used as test sets. The full point set was determined from a still image of the painting, and the standard matching technique from [12] was used for comparison.

We tested the PCA-SIFT descriptor [7] and the SIFT descriptor, both using interest points found with Lowe's Difference of Gaussian detector [12], though we do not believe this technique to be limited to these choices. All sets of query images were labeled by hand for scoring purposes to indicate where the painting is found in each image.

The training set images were matched against the feature points from the still image of the painting and all valid matches (determined by whether a pose hypothesis was formed) recorded and the stability of the reference points determined as described in §2. Then, 10 sets of experiments were run on the test sets with database entries consisting of 500 to 50 stable data points (in increments of 50) to determine how performance degrades as the number of points used for matching decreases. The points are matched using the standard methodology, with the only difference being the reduction of data points to those we have determined to be most stable.

The experimental results were evaluated using a precision/recall curve, with the reported result being the equal error rate for a particular experiment. True positives were determined as query images in which the labeled center of the

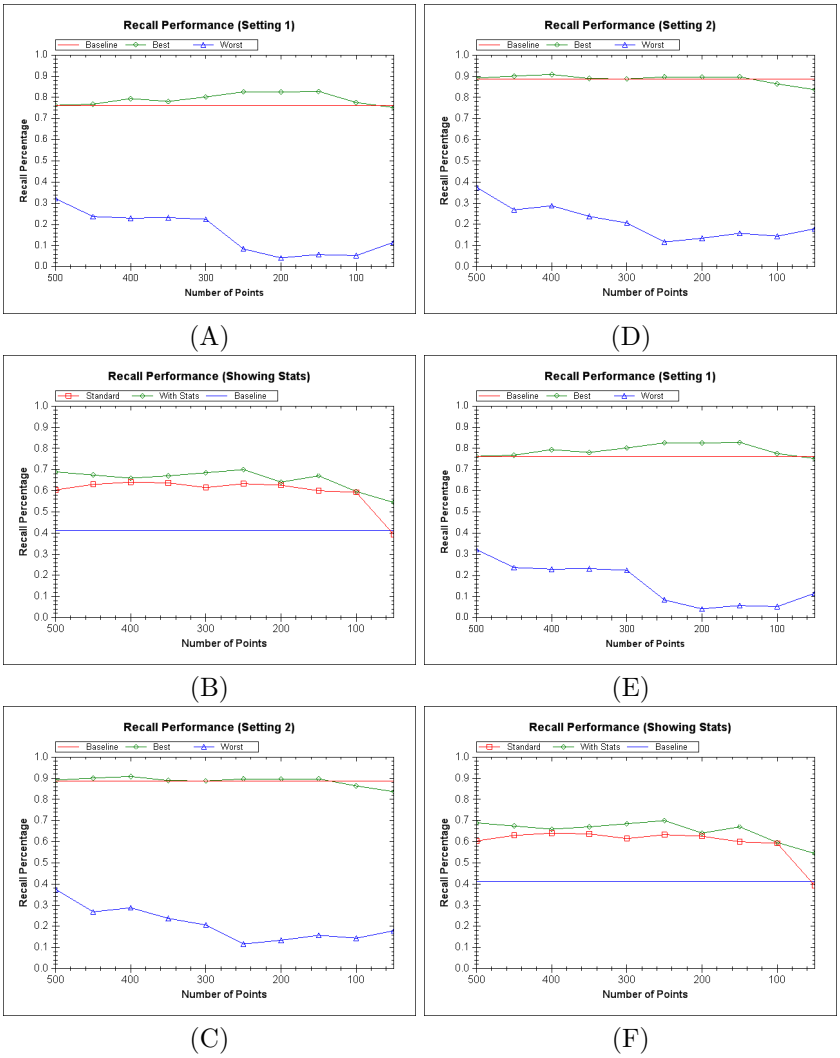

Figure 4: *Results.* In (A) and (B) is shown the recall performance for Stubb's *Gimcrack with John Pratt up on Newmarket Heath* in settings 3 and 4, respectively and in (D) through (E) is shown the same for Titian's *Tarquin and Lucretia*. Results for both the best and worst n points are shown to show the difference a guided subset makes on performance, with the point plotted being the recall at the break-even point. The line indicates the baseline performance using all the points. In (E) and (F) are shown precision/recall curves for database retrieval with the SIFT descriptor and PCA-SIFT descriptor, respectively. One line shows the performance on the painting with the reduced database, the second line with a full database.

reference image in the query lies in the center of the projected image outline in that query. The scoring of each query image that is used for the precision/recall ranking is determined as described in §3. The baseline value shown on the graph is the equal error rate taken from the precision/recall curve resulting from performing a full match with every keypoint in the still image of the painting at each frame. The results can be seen in Figure 4. The performance for the worse n points is shown to underline the fact that just choosing any subset of the interest points is not sufficient.

In addition to individual matching with paintings, we tested the technique when the stable points are stored in a database. For each painting, the 150 most stable points were stored in a database and then retrieval performed on all of the test images and precision/recall curves constructed for each, compared against the curves found when using a similar database with all of the points. Again, we find that although the number of points stored is an order of magnitude less (which, coincidentally, also improves database performance) the performance does not degrade.

As is readily apparent, restricting matching to a subset of stable points not only maintains baseline performance but indeed improves upon it in many cases. In all experiments it was found that one had to reduce the number of points to less than an order of magnitude of the original before performance reduction was observed. The fact that using a reduced subset of stable points improves performance may come as a surprise, however it must be understood that many of the unnecessary points in the full set actually reduce performance by creating false, seemingly coherent poses and otherwise allowing more room for error due to matching with points in the background. Since the stable points have been shown to repeatedly be produced mainly by the object itself as opposed to the background, as we reduce the total number of points using stability the likelihood of this kind of clutter decreases and matching/retrieval performance improves, though at some point (around 50) vital points start to be lost and performance degrades.

5 Conclusion

We have shown that by using our stability measure, it is possible to reduce the number of required database points for a reference image by an order of magnitude while not sacrificing performance. In addition, we introduced a statistical distance metric which further improves performance with small sets of database points. We are interested in exploring further the effect of the training data on performance. We anticipate that there is a performance gain from tailoring a database to a particular application through the use of targeted training data, but would like to determine the extent of the gain. Finally, we have noticed that our stable feature points appear and disappear in what seem to be clearly delineated groups, and are intrigued by the possibility of programmatically determining these groupings, which could be used to perform part-based detection and thus allow for better performance on deformable objects.

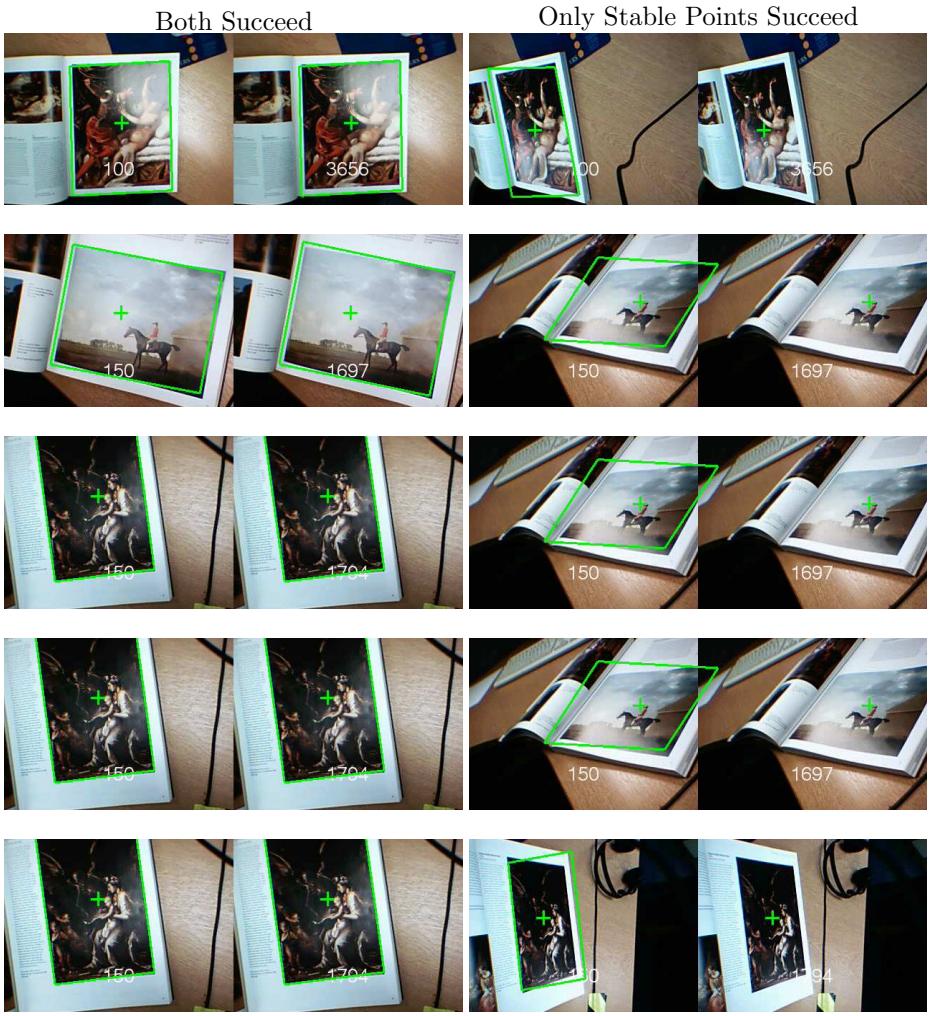

Figure 5: *Query Image Results.* In these images, a side by side comparison of experiments from setting 3 for three objects is shown. In all of them, the image on the left comes from a video made with stable points and on the right from a video made using all the points extracted from a high resolution still image of the object (the number of points used is indicated by the white number at the bottom of the frame). The top three images show frames where both methods recognize the object correctly, and the bottom three show where only stable points achieve a correct pose estimation.

References

[1] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[2] R. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[3] Jason J. Corso and Gregory D. Hager. Coherent regions for concise and stable image description. In *Proceedings of CVPR '05*, 2005.

[4] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[5] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Symposium on Theory of Computing*, pages 604–613, 1998.

[6] Timor Kadir. *Scale, Saliency, and Scene Description*. PhD thesis, University of Oxford, 2002.

[7] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *Proceedings of ACM Multimedia*, 2004.

[8] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of CVPR '04*, 2004.

[9] Svetlana Lazebnik, C. Schmid, and Jean Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of ICCV '03*, pages 649–656, 2003.

[10] V. Lepetit, J. Piley, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of CVPR '04*, Washington, DC, June 2004.

[11] Ting Liu, Andrew Moore, Alexander Gray, and Ke Yang. An investigation of practical approximate nearest neighbor algorithms. In *Proceedings of NIPS '04*, December 2004.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of BMVC '02*, 2002.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of CVPR '03*, June 2003.

[15] S. Obdržálek and J. Matas. Sub-linear indexing for large scale object recognition. In *Proceedings of BMVC '05*, 2005.

[16] S.M. Omohundro. Efficient algorithms with neural network behaviour. *Journal of Complex Systems*, 1(2):273–347, 1987.

[17] F. P. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[18] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[19] Edgar Seemann, Bastian Leibe, Krystian Mikolajczyk, and Bernt Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of BMVC '05*, 2005.

[20] T. Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.