Real-time pose invariant logo and pattern detection
Oliver Sidla, Michal Kottmann, and Wanda Benesova, Proc. SPIE 7878, 78780C (2011), DOI:10.1117/12.872977

http://dx.doi.org/10.1117/12.872977

Text of the paper - see the next side....

Real-time pose invariant logo and pattern detection

Oliver Sidla*[1a], Michal Kottmann[b], Wanda Benesova[b],
[a]SLR Engineering, 8020 Graz, Austria;
[b]Slovak University of Technology, Bratislava, Faculty of Informatics and Information Technologies

ABSTRACT

The detection of pose invariant planar patterns has many practical applications in computer vision and surveillance systems. The recognition of company logos is used in market studies to examine the visibility and frequency of logos in advertisement. Danger signs on vehicles could be detected to trigger warning systems in tunnels, or brand detection on transport vehicles can be used to count company-specific traffic. We present the results of a study on planar pattern detection which is based on keypoint detection and matching of distortion invariant 2d feature descriptors. Specifically we look at the keypoint detectors of type: i) Lowe's DoG approximation from the SURF algorithm, ii) the Harris Corner Detector, iii) the FAST Corner Detector and iv) Lepetit's keypoint detector. Our study then compares the feature descriptors SURF and compact signatures based on Random Ferns: we use 3 sets of sample images to detect and match 3 logos of different structure to find out which combinations of keypoint detector/feature descriptors work well.
A real-world test tries to detect vehicles with a distinctive logo in an outdoor environment under realistic lighting and weather conditions: a camera was mounted on a suitable location for observing the entrance to a parking area so that incoming vehicles could be monitored. In this 2 hour long recording we can successfully detect a specific company logo without false positives.

Keywords: keypoint detection, feature descriptors, logo detection

1. INTRODUCTION

The detection of pose invariant planar patterns has many practical applications in computer vision and surveillance systems. They range from the original application of wide baseline stereo matching to the recognition of company logos in market studies and to the analysis of the visibility and frequency of brand names in advertisement. In traffic monitoring applications vehicles can be detected based on distinctive signs or company logos in order to allow the opening of gates or to count frequencies of incoming company-specific traffic. Danger signs on vehicles could be detected to trigger warning systems or to turn on monitoring equipment in tunnels.

For the purpose of wide baseline point feature matching, a powerful set of new methodologies has been developed over the last years, but the broad usability of these new types of descriptors has enabled many other new applications. These new approaches for interest point detectors and descriptors are nowadays applied not only for wide-baseline stereo matching, but also for pattern matching, image recognition, tracking and navigation, image retrieval and calibration. We present the results of a study on logo detection and matching which is based on the localization of robust image keypoints and their description by means of scale-, rotation-, and transformation-invariant 2d features. Available keypoint detectors differ in great amounts by their offline pre-detection efforts, their memory requirements and their runtime speed, so do the available feature description methods. This study is aimed at an analysis of how well different keypoint detector and descriptor combinations work in practice. To this end we have designed two different tests. The first set of experiments uses 3 different logo types, one taken from publicly available videos, two are generated by the authors. They are used to examine the limits of the 2d feature descriptors under variations of pose, perspective, resolution and image quality. The second type of experiments, as a real-world test, tries to detect vehicles with a distinctive logo in an outdoor environment: a camera was mounted at the entry to a parking area so that it could observe all incoming and outgoing vehicles. The recorded image sequence used for this test spans about two hours.

[1] os@slr-engineering.at; +43 (0) 699 11874796; www.slr-engineering.at

The rest of the paper is structured as follows: Section 2 gives an overview of the state-of-the-art on keypoint detection and feature descriptors with emphasis on the relevant algorithms used in this work, Section 3 then explains our testing procedure, it also details our logo detection algorithm, and Section 4 presents the testing results, including the recognition results from our real-world test. Section 5 summarizes our findings and gives, with our type of application in mind, a recommendation of a keypoint/descriptor combination which has the best speed/accuracy/robustness tradeoff.

2. STATE OF THE ART

The detection of salient image points and their description in terms of local properties or features is an important task useful for many applications like stereo matching and tracking. A characterization respective description of a local point neighborhood in terms of invariant features becomes even more important for stereo matching when the camera baseline increases and more distortion is introduced between images of the same local image patch. In this case similarity measures like local gray value correlation will often fail because skewing and rotation can not be handled by these simple algorithms anymore.

Lowe has introduced the SIFT operator in 1999, a complete and updated complete reference can be found in his 2004 journal paper[1]. Lowe first searches for keypoints using a difference of Gaussians approach in several resolution scales. Local maxima in the scale space define stable keypoints including their actual scale in the image. Around those keypoints edge orientation histograms are used to determine the basic reference orientation of the keypoint. The keypoint descriptor then computes local histogram features relative to this keypoint orientation. Lowes operator is still the standard today in terms of robustness and recognition rate, it has the disadvantage though that it is expensive from a computational point of view and hardly real-time capable on low performance CPU architectures.

Bay et. al[2] improved on the SIFT algorithm by modifying the interest point detection approach, they further sped up and approximated the point detection process with the help of integral images. By simplifying all steps of the point description algorithm, their SURF detector becomes real-time capable, with a detection performance comparable to the original SIFT algorithm.

Rosten and Drummond[9] recognize the relative slow speed of the Harris[11], SIFT[1], or SUSAN[14] keypoint detectors. In their work they treat keypoint detection as a machine learning problem. By learning and creating decision trees, they come up with an algorithm which can not only run very efficiently but also has a high repeatability equal or better than other corner detection approaches (a fact that we have also observed in this study).

Mikolajczyk et al.[11] analyze very thoroughly several affine region detectors and try to establish a reference set of images and software, they extend their analysis to descriptors and introduce GLOH, another very well performing local feature descriptor in their subsequent work[12].

A new paradigm for point detection and especially point description has been introduced by Lepetit and Fua[3]. They treat point matching essentially as a classification task by using a very simple descriptor and training a classifier offline from many artificial samples of possible keypoints[4,5]. In their run-time system they use a classification tree to decide whether a local image region, as described by gray value comparisons, is actually one of the trained keypoint patches. By offloading some of the processing work to a one-time preprocessing stage, Lepetit and Fua can make their keypoint detector real-time capable. Still the training of new possible image patches which need to be recognized is a time consuming procedure and may take at least several seconds. Re-training or the addition of new image points is therefore not feasible in a real-time system with this approach.

M. Özuysal et. al.[8] improve the original Randomized Tree classifier of Lepetit by their Random Ferns approach. The deficiency of a relatively time consuming offline training procedure is attacked by Calonder et. at. who uses a very similar description scheme as Lepetit, but adds the capability to learn new keypoint descriptors online within their algorithm[6]. Their idea is to use the offline training with Random Ferns as starting point and then describe new keypoints by the response vector of the previously learned Ferns on the new image patch. Keypoint recognition is then a matter of comparing the responses of detected points to stored (= trained) response vectors. This training as well as the recognition process is real-time capable. Calonder et. al. show that the discriminative power of their keypoint descriptor is comparable to SIFT.

By further compressing their signature vectors using a projection matrix (this can either be PCA or a random projection matrix) Calonder et. al. can very significantly reduce the memory footprint of their approach and speed it up as well[7]. They claim that their algorithm is about 30 times faster than SIFT and about 4 times faster than their original signature algorithm.

3. TEST METHODOLOGY

3.1 Overview

Our tests are aimed at examining the performance of different keypoint detection and description methods in order to find good and reasonable combinations which can be used in real-world applications. Specifically we detect keypoint locations with one method of

- DoG approximation which is used in the SURF[2] algorithm (SURF)
- Harris corner detection[13] (HARRIS)
- FAST[10] corner detector (FAST)
- Leptit keypoint detector[3,6] (LDETECT)

As image descriptors we have tested

- SURF[2] (SURF)
- Calonders signature based on random Ferns[6,7] (FERNS)

The matrix in Table 1 shows the detector/classifier combinations which have been tested in our experiments. In the rest of this work we describe the keypoint detector method and feature descriptor method abbreviated in the form *keypoint/descriptor,* for example FAST/SURF means that we use the FAST corner algorithm as keypoint detector and the SURF method as feature descriptor.

The detector/classifier combinations of Table 1 have been tested on 3 datasets from different sources and of different pattern type. Figure 2 shows the reference images which have been used for testing. The patterns were chosen so that different texture characteristics (corners, curves) could emphasize possible advantages or disadvantages of the keypoint detectors and the feature descriptors: FEDEX consists mainly of corners and straight lines, FORD is very rounded, BOX has a lot of structure and texture, and DHL is a mixture of straight and rounded features. Figure 3 shows typical images from the 3 datasets FEDEX, FORD, and BOX.

The pattern DHL has been used in a dataset which we have created for a more realistic real-world test. The DHL logo has been trained and then detection was run on a long outdoor video sequence which spans a time of about 2 hours. This test mainly allowed us to estimate the false positive rate of our detection pipeline.

Table 1: The different keypoint/descriptor combinations, marked by a •, which have been used in our tests.

		keypoint detection method			
		HARRIS	**LDETECT**	**SURF**	**FAST**
keypoint descriptor	**SURF**	•		•	•
	FERNS	•	•	•	•

FEDEX	FORD	BOX	DHL

Figure 2: The 4 logos which have been used for our artificial tests and the pattern used in the matching test on the outdoor video (DHL). FEDEX has a pattern with corners and no curvature, FORD is a very rounded pattern, and BOX has a lot of texture and possibly many keypoint candidates.

Figure 3: Typical images from the test set for FEDEX (left), FORD (middle) and BOX (right). The images from the FEDEX and FORD dataset do have a poor quality (low contrast, noise, reflections).

3.2 Pattern matching strategy

For the evaluation of keypoint detectors and feature descriptors we have mainly relied on their implementation in the OpenCV computer vision library. Around the basic keypoint/descriptor functions we have built our detection algorithm which is outlined in Algorithm 1. After keypoint computation and feature generation we make use of the fast nearest neighbor search method which is based on kD-trees[1]. Although kD-trees are less efficient with high dimensional features, they still can speed up the nearest-neighbor search for matching significantly.

The logo detection algorithm first finds possible keypoints and their descriptors in the input image. The pre-computed logo keypoints are then being matched to the image keypoints by making use of Lowe's observation that for a valid pattern point to match, its feature distance to the best matched image keypoint must be significantly smaller then the feature distance to the second best match[1]. By collecting only valid matches we then construct a homography from the image points to the logo image space. If this homography is plausible and the resulting polygon of the logo's 4 corner points is convex, we assume that we have found the logo in the input image. By removing the matched image keypoints and repeating the procedure, we can search for another occurrence of the logo so that more than one instance of it can be detected.

Algorithm 1 describes also the learning and detection process as it has been implemented for this study. The learning procedure consists simply of creating the keypoint locations and descriptors for the logo and storing them as reference. Only for the LDETECT and FERN algorithms additional preprocessing is necessary to train the keypoint detector and the descriptor's Random Ferns classifier

Algorithm 1: Logo pattern training, logo pattern detection. The learning procedure consists simply of creating the keypoint locations and descriptors of the logo. Only for the LDETECT and FERN algorithms an additional preprocessing step is necessary.

logo learning

1. train keypoint detector for method LDETECT
2. train descriptor for method FERN
3. use the keypoint detector to find logo keypoints PL_l
4. calculate the descriptors for the keypoints FDL_l

logo matching

do {

1. empty set of accepted matches **A**
2. use keypoint detector to find image keypoints PI_i
3. calculate the descriptor set **FDI** for each PI_i
4. build kD-Tree structure from **FDI**
5. for each logo descriptor FDL_l :
    a. find the best and second best match in **FDL** with distances D_1 and D_2
    b. if $D_1 < 0.6*D_2$, keep FDL_l and add it to the set of accepted matches **A**
6. filter out duplicates from **A** - only one-one matched pairs are accepted
7. if (size(**A**) < 4) match = false; exit;
8. calculate homography from all points in **A** using RANSAC to select valid logo keypoints
    a. determine the affine transformation of the logo into the image
9. plausibility test of calculated homography by testing the logo corners and testing of convexity:
    a. transformed logo corners have to be inside the image,
    b. polygon of transformed logo boundaries have to be convex
    c. if 9a) and 9b) are accepted, match = ***true***; else match = ***false***;
10. Remove points in **A** from the set of image keypoints **FDI** and continue with step 1

} until (match == ***false***)

4. TEST RESULTS

4.1 Tests on artificial image sequences

Algorithm 1 has been used to test the different keypoint/descriptor combinations from Table 1 on the three image data sets which are listed again in Table 2. The following Tables 3, 4, and 5 give the detection results for these datasets.

The results from the tests on the artificial datasets show that the combination of the original SURF keypoint detector and the SURF feature descriptor seem to outperform all other tested keypoint/descriptor combinations except for the FORD test, where it is beaten by the LDETECTOR/FERN combination by a small margin. The LDETECTOR algorithm seems to work better on the FORD logo possibly because of the rounded pattern structure. It does not depend strictly on good edges or corners like the rest of the keypoint detectors, especially HARRIS and FAST. The HARRIS corner detector seems to be the least favorable in terms of keypoint detection. Obviously its repeatability is not good enough for that purpose. The FAST corner detector does perform better, but it is by far not the best keypoint detector. One of its advantages is that its run-time performance beats all other algorithms by an order of magnitude.

The results of our tests on the 3 datasets shows that the combination of the SURF keypoint detector and the SURF feature descriptor currently seems to be the best compromise in terms of robustness and speed. The LDETECTOR/FERN combination did not perform particularly well, at least for our type of application.

Table 2: The 3 datasets and the number of images which they contain. The dataset FEDEX has been taken from Youtube videos (search for FedEx), datasets 2 and 3 have been generated by the authors.

Dataset	**Nr Images**
FEDEX	101
FORD	19
BOX	12

Table 3: The different keypoint/descriptor combinations and their detection rate on the FEDEX dataset. The best results is marked in boldface.

		keypoint detection method			
		HARRIS	**LDETECT**	**SURF**	**FAST**
keypoint descriptor	**SURF**	10.9%		**60.4%**	30.7%
	FERNS	0%	25.7%	0%	1.99%

Table 4: The different keypoint/descriptor combinations and their detection rate on the FORD dataset.

		keypoint detection method			
		HARRIS	**LDETECT**	**SURF**	**FAST**
keypoint descriptor	**SURF**	0%		26.3%	5.3%
	FERNS	10.5%	**31.6%**	0%	10.5%

Table 5: The different keypoint/descriptor combinations and their detection rate on the BOX dataset.

		keypoint detection method			
		HARRIS	**LDETECT**	**SURF**	**FAST**
keypoint descriptor	**SURF**	0%		**83.3%**	50.0%
	FERNS	41.7%	75.0%	8.3%	66.7%

4.2 Pattern matching in the outdoor video sequence

The tests in Section 4.1 have shown that the SURF/SURF logo detection method seems to be the most reliable approach at an acceptable level of run-time performance. For a further test of our logo detection, as described in Algorithm 1, we have taken a sequence of images from a camera which has been located at the entrance of a parking area for the duration of about 2 hours. The camera recorded images at a rate of 4 frames/sec. In this sequence we have tried to detect the DHL logo with the both the SURF/SURF and the LDETECT/FERN keypoint/descriptor combination under real-world conditions in images of low quality, similar to what can be found in many outdoor surveillance videos.

As it is shown in Figure 4, the SURF/SURF combination has detected the DHL logo in several frames (top row in Figure 4). For a comparison, the lower row of images in Figure 4 shows frames where the detection of the DHL logo was just lost (left and middle image) and an image where another vehicle enters the parking area. This image in the lower right side is typical for most of the other frames – there are a few keypoint matches but they are not sufficient to trigger detection of the pattern. The DHL logo has not (falsely) been detected within the sequence of test images other than on the correct occasion when the DHL van left the area (the recording was started as the van was already in the parking area, so its entry has not been captured by the camera). This result means that the false positive rate of the SURF/SURF logo detector is low enough to be practical for real-world applications.

The LDETECT/FERN combination did not detect the DHL logo correctly.

Figure 4: The SURF/SURF logo detection on the outdoor sequence at the parking area.
The DHL logo has been correctly detected, no false matches occurred.

5. CONCLUSION

In this work we have tested several keypoint/feature descriptor combinations for logo detection in order to estimate the robustness of different algorithms with respect to logo type and image quality. Our tests have shown that the well established SURF/SURF combination seems to perform best, followed by Calonder’s keypoint detector/Random Fern combination. The authors still believe that the LDETECTOR/Fern combination, especially when using the compressed signature framework is used (which has not yet been implemented for this test) bears a large potential. Its speed advantage and ability for fast online learning should make it an interesting alternative to SURF/SURF.

The SURF/SURF algorithm performed well even in the low quality images of our outdoor test sequence, we believe therefore that for the purpose of logo/pattern detection SURF still seems to be the best currently available option.

ACKNOWLEDGEMENTS

This work was supported by grant KEGA 244-022STU-4/2010.

REFERENCES

1. D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, 60(2):91– 110 (2004).
2. H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359 (2008).
3. V. Lepetit and P. Fua, “Towards Recognizing Feature Points Using Classification Trees”, Technical Report IC/2004/74, EPFL (2004).
4. V. Lepetit, P. Lagger, P. Fua, “Randomized Trees for Real-Time Keypoint Recognition”, Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, CA, June 2005.
5. V. Lepetit, P. Fua, “Keypoint Recognition using Randomized Trees”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, Nr. 9, pp. 1465-1479 (2006).
6. M. Calonder, V. Lepetit, and P. Fua, “Keypoint Signatures for Fast Learning and Recognition”, Proceedings of European Conference on Computer Vision (2008).
7. M. Calonder, V. Lepetit, P. Fua, K. Konolige, J. Bowman, and P. Mihelich, “Compact Signatures for High-speed Interest Point Description and Matching”, Proc. of the International Conference on Computer Vision, 2009.
8. M. Özuysal, M. Calonder, V. Lepetit, P. Fua, “Fast Keypoint Recognition using Random Ferns”, IEEE Transactions on Pattern Analysis and Machine Intelligence (2010).
9. E. Rosten, T. Drummond, "Machine learning for high-speed corner detection", Proc. European Conference on Computer Vision, Volume 1, pp.440-443 (2006).
10. C. Harris and M. Stephens, “A combined corner and edge detector”, Proc. of the 4th Alvey Vision Conference, pp. 147–151 (1988).
11. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, “A comparison of affine region detectors”, International Journal of Computer Vision, 65(1/2):43–72 (2005).
12. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615-1630 (2005).
13. B. Peleato, M. Jones, “Faster-than-SIFT Object detection”, Stanford University Course Project CS223 report, 2010.
14. S. M. Smith, J. M. Brady, “SUSAN - a new approach to low level image processing,” International Journal of Computer Vision, vol. 23, no. 1, pp. 45–78 (1997).