Training for Task Specific Keypoint Detection

Christoph Strecha, Albrecht Lindner, Karim Ali and Pascal Fua

CVLab EPFL
Lausanne Switzerland

Abstract. In this paper, we show that a better performance can be achieved by training a keypoint detector to only find those points that are suitable to the needs of the given task. We demonstrate our approach in an urban environment, where the keypoint detector should focus on stable man-made structures and ignore objects that undergo natural changes such as vegetation and clouds. We use WaldBoost learning with task specific training samples in order to train a keypoint detector with this capability. We show that our aproach generalizes to a broad class of problems where the task is known beforehand.

1 Introduction

State of the art keypoint *descriptors* such as SIFT [1] or SURF [2] are designed to be insensitive to both perspective distortion and illumination changes, which allows for images obtained from different viewpoints and under different lighting conditions to be successfully matched. This capability is hindered by the fact that general-purpose keypoint *detectors* exhibit a performance which deteriorates with seasonal changes and variations in lighting. A standard approach to coping with this difficulty is to set the parameters of the detectors so that a far greater number of keypoints than necessary are identified, in the hope that enough will be found consistently across multiple images. This method, however, entails performing unnecessary computations and increases the chances of mismatches.

In this paper, we show that when training data is available for a specific task , we can do better by training a keypoint detector to only identify those points that are relevant to the needs of the given task. We demonstrate our approach in an urban environment where the detector should focus on stable man-made structures and ignore the surrounding vegetation, the sky and the various shadows, all of which display features that do not persist with seasonal and lighting changes. We rely on WaldBoost learning [3], similar in essence to the recent work [4] by the same authors, to learn a classifier that responds more frequently on stable structures.

Task-specific keypoint detection is known to play an important role in human perception. Among the early seminal studies is that of Yarbus [5] where it was demonstrated that a subject's gaze is drawn to relevant aspects of a scene and that eye movements are highly influenced by the assigned task, for instance memorization. To the best of our knowledge, these ideas have not yet made their mark for image-matching purposes. Our main contribution is to show that image matching algorithms benefit from incorporating task-specific keypoint detection.

We begin this paper with a brief review of related approaches. Next, we discuss in more detail what constitutes a stable keypoint that an optimized detector should identify and introduce our approach to training such a detector. Experimental results are then presented for the structure and motion problem, where our goal is to build a keypoint detector - called TaSK (Task Specific Keypoint) that focuses on stable man-made structure. We also show a result of a keypoint detector, which was learned to focus on face features. Finally, we conclude with a discussion.

2 Related work

State of the art keypoint detectors fall into two broad categories: those that are designed to detect corners on one hand, and those that detect blob-like image structures on the other. An extensive overview can be found in Tuytelaars *et al.* [6]. Corner like detectors such as Harris, FAST [7], Förstner [8] [9, 10] are often used for the pose and image localization problems. These detectors have a high spatial precision in the image plane but are not scale invariant and are therefore used for small baseline matching or tracking. The other category of keypoint detectors aim at detecting blob structures (SIFT [1], MSER [11] or SURF [2]). They provide a scale estimate, which renders them suited for wide-baseline matching [12, 13] or for the purpose of object detection and categorization. Both detector types can be seen as general-purpose hand crafted detectors, which run for many application at a very high false positive rate to prevent failures from missed keypoints.

Our approach is most related to the work of Šochman and Matas [4]. These authors, emulate the behavior of a keypoint detector using the boosting learning method. They show that the emulated detector achieves equivalent performance with a substantial speed improvement. Rosten and Drummond [7, 14] applied a similar idea to make fast decisions about the presence of a keypoints in a image patch. There, learning techniques are used to enhance the detection speed for general-purpose keypoint detection. Note, that their work does not focus on task specific keypoint detection, which is the aim of this paper. Similar in spirit is also the work of Kienzle *et.al.* [15] in which human eye movement data is used to to train a saliency detector.

3 Task specific keypoints

Training data can be used in various ways to improve the keypoint detection. We will describe two approaches in the following sections.

3.1 Detector verification

Suppose we are given a keypoint detector $\mathcal{K}$ and a specific task for which training data is available. The most natural way to enhance keypoint detection is based on a post-filtering process: among all detections which are output by the detector $\mathcal{K}$, we are interested only in the keypoints that are relevant given the training data. Our enhanced keypoint detector would then output all low-level keypoints and an additional classification stage is added which rejects unreliable keypoints based on the learned appearance.

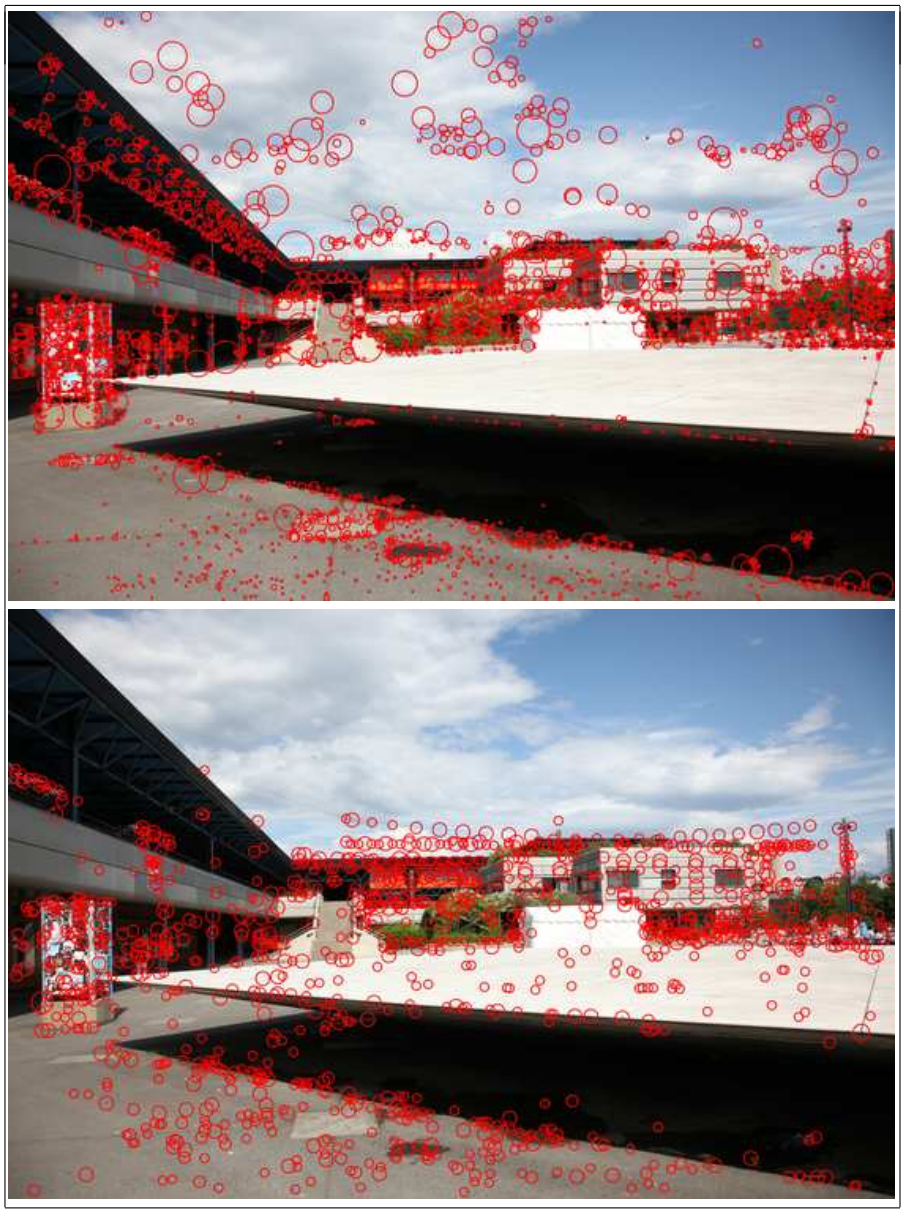

Fig. 1. Keypoint detections by DoG (top) and our proposed detector TaSK (bottom). Note that TaSK is specialized to focus more on stable man-made structures and ignores vegetation and sky features.

3.2 Detector learning

In order to learn the appearance of good keypoints we need to specify how they are characterized. In particular we need to specify the conditions under which a pixel can be regarded as a good keypoint. We will use the following two criteria:

1. A good keypoint can be reliably matched over many images.
2. A good keypoint is well localized, meaning its descriptor is sufficiently different from the descriptors of its neighboring pixels.

All pixels that obey these criteria will constitute the positive input class to our learning while the negative training examples are random samples of the training images.

Our method is based on WaldBoost learning [3] similar in spirit to the work of Šochman and Matas [4]. Using our aforementioned training examples, we learn a classifier that responds more frequently on stable structures such as buildings and ignores unstable one such as vegetation or shadows. Our eventual goal is to only detect keypoints that can be reliably matched. The advantage is not only a better registration, but also a speed up in the calibration.

For the WaldBoost training we used images taken by a panorama camera. These images are taken from the same view point every 10 minutes for the past four years. This massive training set captures light and seasonal changes but does not cover appearance variations which are due to changes in view point.

3.3 Training samples

The generation of the training samples is an important preliminary step for the detector learning since the boosting algorithm optimizes for the provided training samples. In [3], the set of training samples fed into the boosting algorithm is the set of *all* keypoints identified by a specific detector. In so doing, the learned detector is naturally no more than an emulation of the detector for the training samples.

Our research aims at generating a more narrow set of training samples, which obey the criteria proposed in section 3.2. In a first step, we used the Förstner [8] operator to find keypoint candidates which are well localized in the images. In a second step, keypoints which are estimated to have poor reliability for reconstruction purposes are pruned.

The automated selection of keypoints is based on two features: the number of occurrences of a keypoint and the stability of a descriptor at a specific position over several images of the sequences.

The number of occurrences is simply a count of how many times a fixed pixel position has been detected as a keypoint in several images of the same scene. To illustrate our measure of stability, let p_i^j denote the position of the i-th keypoint in the j-th image $i = 1 \dots N_j, j = 1 \dots N_{images}$. The union $\mathrm{P} = \bigcup p_i^j$ contains all the positions which have been detected in at least one image. In all the images a SIFT descriptor s_k^j is calculated for every single position $p_k \in \mathrm{P}$. For the stability of the descriptor Euclidean distances $d_k^{j_1,j_2} = \mathrm{dist}(s_k^{j_1}, s_k^{j_2})$ are calculated and their median $d_k = median(d_k^{j_1,j_2}), j_1 \neq j_2$ is determined. The more stable a keypoint is in time, the smaller its median will be. A pixel position is then classified as a good keypoint if its occurrence count is high and its descriptor median is low: two thresholds were thus set so that a reasonable number of keypoints is obtained for our training set(couple of thousands per image). These keypoints form the positive training set. The negative training examples are randomly sampled from the same images such that they are no

closer than 5 pixels to any positive keypoint. Given these training examples we apply WaldBoost learning, as described in the next section.

4 Keypoint boosting

Boosting works by sequentially applying a, usually, weak classification algorithm to a re-weighted set of training examples [16, 17]. Given N training examples $x_1 \dots x_N$ together with their corresponding labels $y_1 \dots y_N$, it is a greedy algorithm which leads to a classifier $H(x)$ of the form:

$$H(x) = \sum_{t=1}^{T} h_t(x) \ , \tag{1}$$

where $h_t(x) \in \mathcal{H}$ is a weak classifier from a pool $\mathcal{H}$ chosen to be simple and efficient to compute. $H(x)$ is obtained sequentially by finding at each iteration t the weak classifier which minimizes the weighted $D_t(x_i)$ training error:

$$Z_t = \sum_{x_i=1}^{N} D_t(x_i) \exp(-y_i h_t(x_i)) \ . \tag{2}$$

The weights of each training sample, $D_t(x_i)$, are initialized uniformly and updated according to the classification performance. One possibility to minimize eq. 2 uses domain partitioning [17] as next explained.

4.1 Fuzzy weak learning by domain-partitioning

The minimization of eq. 2 includes the optimization over possible features with response function $r(x)$ *and* over the partitioning of the feature response into $k = 1 \dots K$, non-uniformly distributed bins. If a sample point x falls into the k^{th} bin, its corresponding weak classification result is approximated by c_k. This corresponds to the real version of AdaBoost.[1] By this partitioning model, eq. 2 can be written as (for the current state of training t):

$$Z = \sum_{k=1}^{K} \sum_{r(x_i) \in k} D(x_i) \exp(-y_i c_k) \ . \tag{3}$$

To compute the optimal weak classifier for a given distribution $D(x_i)$ many features r are sampled and the best , *i.e.* the one with minimal Z is kept.

The optimal partitioning is obtained by rewriting eq. 3 for positive ($y_i = 1$) and negative ($y_i = -1$) training data:

$$Z = \sum_{k=1}^{K} \left(W_k^+ \exp(-c_k) + W_k^- \exp(c_k) \right) \ ,$$

[1] For the discrete AdaBoost algorithm, a weak classifier estimates one threshold t_0 and outputs $\alpha = \{-1, 1\}$ depending of whether a data point is below or above this threshold.

ALGORITHM: WaldBoost Keypoint learning

Input: $h \in \mathcal{H}, (x_1, y_1) \dots (x_1, y_1), \theta^+, \theta^-$
initialize weights $D(x_i) = 1/N$; mark all training examples as undecided $\{y_i^* = 0\}$
For $t = 1 \dots T$, number weak learners in cascade
 sample training examples x_i from undecided examples $\{y_i^* = 0\}$
 compute weights $D(x_i)$ w.r.t. $H_{t-i} \forall \{y_i^* = 0\}$
 For $s = 1 \dots S$, number weak learner trials
  -sample weak learner $h_t \in \mathcal{H}$
  -compute response $r(x_i)$
  -compute domain partitioning and score Z [17]
 End
 -among the S weak learners keep the best and add h_t to the strong classifier $H_T = \sum_t h_t$
 -sequential probability ratio test[3]
 classify all current training examples into $y_i^* = \{+1, -1, 0\}$
End

Fig. 2. WaldBoost Keypoint learning

where

$$W_k^{+/-} = \sum_{r(x_i) \in k} D_k^{+/-}(x_i) \tag{4}$$

is the sum of positive and negative weights $D_k^{+/-}$ that fall into a certain bin k.

After finding the optimal weak learner, Wald's decision criterion is used to classify the training samples into $\{+1, -1, 0\}$ while the next weak learner is obtained by only using the undecided, zero labelled, training examples. The entire algorithm is shown in table 4.1. For more information we refer to the work of Schapire *et.al.* [17].

4.2 Weak classifier

The image features which are used for the weak classifiers are computed by using integral images and include color as well as gradient features. For the minimization of 4, we first randomly sample a specific kind of weak classifier and than its parameters. The weak classifiers include:

- ratio of the mean colors of two rectangles: compares two color components of two rectangles at two different positions ($2+4+4$ parameters).
- mean color of a rectangles: measures the mean color components of a rectangles ($1+2$ parameters).
- roundness and intensity: integral images are computed from the componnet of the structure tensor, roundness and intensity as defined by Föstner and Gülch [8] are further computed on a randomly sampled rectange size (2 parameters).

5 Detector evaluation

Repeatability is a main criterion for evaluating the performance of keypoint detectors. In contrast to current studies by Mikolajczyk *et al.* [18] where a good feature detection

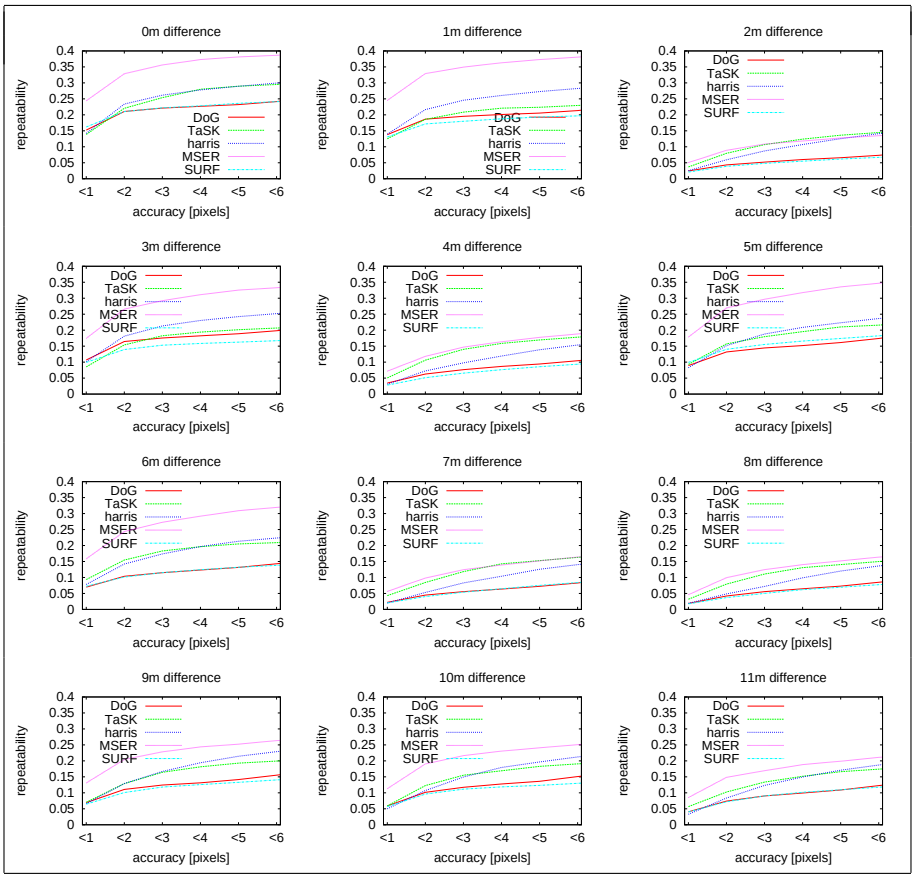

Fig. 3. Repeatability evaluation for seasonal changes. Repeatability scores for matching January with all other months (all images are taken at the same time of the day).

was defined according to the percentage of overlap between the keypoint ellipses, we evaluate repeatability more specifically for the task of image calibration. The Mikolajczyk criterion is in fact not well suited to evaluate multi-view image calibration, where a successful calibration should result in a sub-pixel reprojection error. We are more interested in a keypoint location which only deviates by a few pixels from the ideal keypoint location. Our evaluation is performed as follows: given a reference image, we calculate all keypoints obtained from a specific detector on all images for which the transformation to the reference image is available.

Repeatability is now defined as the percentage of detections in another image that lie within a radius of $n, n = 1 \ldots 6$ pixels. Hence, for every keypoint in the reference image, we perform a search in the target image to identify the closest detection with respect to the ground truth localization. This event is placed in the n^{th} bin of the repeatability,

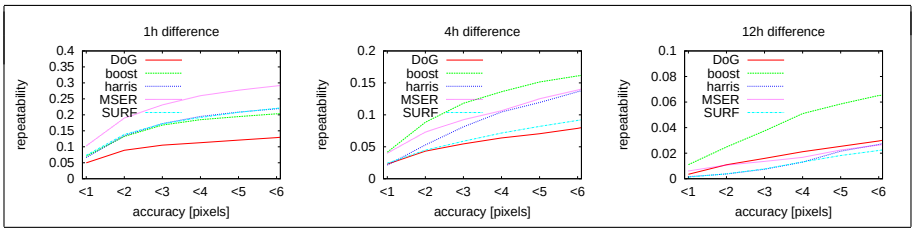

Fig. 4. Repeatability evaluation for daily changes in light. Time difference between the images is 1h (left), 4h (middle) and 12h (right).

while both keypoints are marked as already matched and not considered further. This procedure is repeated until valid keypoints have been assigned to one of the bins.

5.1 Light and seasonal changes

To evaluate the performance with respect to light an seasonal changes we used 65 images of a panorama camera. Images from different times of the day and from different months of the year are used. The set of images thus covers a great variety of lighting conditions such as different incident angles, intensity and inhomogeneity due to cloud coverage. All images are perfectly aligned. The repeatability measures are shown in fig. 3 and fig. 4. On the x-Axis is the accuracy, that is the distance between the closest pair of keypoints from two different images. On the y-Axis is the ratio of the amount of pairs with a certain distance to the total number of keypoints.

The 12 subfigures in fig. 3 show seasonal comparisons. An image from each month has been compared to an image from January. The time difference in months is indicated in the title of each subfigure. Depending on the appearance of the scene in the different months the repeatability varies a lot. It is evident that the time differences of zero and 1 month result in the best repeatability.

The 3 subfigures in fig. 4 show comparisons between images taken at different times of the same day. The time difference in hours is indicated in the title of each subfigure.

From both figures it can be observed that repeatabilities are almost always in the same range. Only in the case of comparisons with different images of the day, the repeatabilities are significantly smaller. This is reasonable since the incident angle of the sunlight changes a lot during the day but much less during the year (recall, all images in fig. 3 have been taken at noon).

In the cases of extreme light changes (fig. 4 middle and right) the TaSK detector outperforms all the other detectors and provides the most reliable keypoint detections under these very difficult conditions. In the less difficult seasonal changes the TaSK detector performs roughly as 2nd best after MSER. The good performance of MSER can be explained by the fact that the test images do not contain geometric transformations.

Additionally we measured how many detected keypoints lie in regions with stable structures (buildings, streets, mountains, ...) and regions with unstable structures (sky, vegetation, ...). Fig. 1 shows that the TaSK detector focuses its detections on stable

Fig. 5. Keypoint detections by DoG (left) and our proposed detector TaSK (right). Note that TaSK is in this case specialized to focus more on face features.

regions with 79% of the total number of keypoints lying in man-made structures, while the DoG detector has less than 59 % of keypoints in those regions.

In fig. 5, we show the detection result of DoG and TaSK of faces. Not that in this case we have trained the TaSK detector on a different set of positive examples. This was selected by takeing keypoint detections on faces as a positive set. Random samples of images which do not contain faces have been choosen to be the negative set.

6 Conclusions

This paper deals with the learning of task specific keypoint detectors (TaSK) by using boosting. Given training examples of good keypoints, we trained a classifier to distinguish the latter from random image patches. This results in a keypoint detector, which produces high repeatability scores on challenging scenes with strong light and seasonal changes.

As an example we trained a keypoint detector to work with higher repeatability on structure and motion applications. For this application, it is often a problem to match images with strong light and seasonal changes. General purpose keypoint detectors usually produce many keypoints on vegetation, which are a-priori known to be ineffectual for matching. Our trained keypoint detector (TaSK) has this knowledge incorporated.

Often and in many applications such as pose estimation, structure from motion, object detection and categorization, general purpose detectors are used. We argued here, that task specific keypoint detectors can increase the performance when tuned to the

specific task, which is often known beforehand. To show this we also included an example on a keypoint detector for faces.

This research was supported by Nokia Reseach Center and Deutsche Telekom Laboratories.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision* **60**(2) (2004) 91–110
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: *Proc. European Conf. on Computer Vision* . (2006) 404–417
3. Šochman, J., Matas, J.: Waldboost - learning for time constrained sequential detection. In Schmid, C., Soatto, S., Tomasi, C., eds.: *Proc. Int'l Conf. on Computer Vision and Pattern Recognition* . Volume 2., Los Alamitos, USA, IEEE Computer Society (June 2005) 150–157
4. Šochman, J., Matas, J.: Learning a fast emulator of a binary decision process. In Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H., eds.: *Proc. Asian Conf. on Computer Vision* . Volume II of LNSC., Berlin Heidelberg, Springer (2007) 236–245
5. Yarbus, A.L.: Eye movements and vision. Plenum. New York (1967 (Originally published in Russian 1962))
6. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Found. Trends. Comput. Graph. Vis. **3**(3) (2008) 177–280
7. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: *Proc. European Conf. on Computer Vision* . Volume 1. (May 2006) 430–443
8. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centers of circular features. In: Proceedings of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data (1987) 281–305
9. Ouellet, J., Hébert, P.: Asn: Image keypoint detection from adaptive shape neighborhood. In: *Proc. European Conf. on Computer Vision* . (2008) 454–467
10. Agrawal, M., Konolige, K., Blas, M.: Censure: Center surround extremas for realtime feature detection and matching. In Forsyth, D.A., Torr, P.H.S., Zisserman, A., eds.: *Proc. European Conf. on Computer Vision* . Volume 5305 of Lecture Notes in Computer Science., Springer (2008) 102–115
11. Matas, J., Chum, O. Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *Proc. British Machine Vision Conf.* . (2002) 384–393
12. Vergauwen, M., Van Gool, L.: Web-based 3d reconstruction service. Mach. Vision Appl. **17**(6) (2006) 411–426
13. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH '06, New York, NY, USA, ACM Press (2006) 835–846
14. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **99**(1) (5555)
15. Kienzle, W., Wichmann, F.A., Schlkopf, B., Franz, M.O.: Learning an interest operator from human eye movements. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop, IEEE Computer Society (04 2006) 24
16. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
17. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. In: Machine Learning. (1999) 80–91

18. Mikolajczyk, K., Tuytelaars.T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. *Int'l Journal of Computer Vision* **65**(1-2) (2005) 43–72