Optimal Number of Image Keypoints for Real Time Visual Odometry

Volker Nannen * **Gabriel Oliver** **

Departament de Ciències Matemàtiques i Informàtica
Universitat de les Illes Balears, 07170 Palma de Mallorca, Spain

* vnannen@gmail.com ** goliver@uib.es

Abstract: A visual odometer can estimate robot motion by tracking a set of invariant keypoints over a sequence of camera frames, at a computational cost that is roughly linear in the number of extracted keypoints. The leading literature suggests to extract all keypoints with a response value—e.g., a Laplacian or Hessian determinant—above a given threshold. We find that the number of image keypoints that pass a given threshold is highly variable between images, which is impractical for real time systems where motion needs to be estimated in constant time.
Here we propose to extract a constant number of keypoints that have the highest response for their respective frame. To find the optimal number of extracted keypoints, we define a range of thresholds on odometer performance, and study how the number of image pairs for which a visual odometer passes the thresholds depends on the number of extracted keypoints. We find that the shapes of the resulting graphs are relatively invariant to the chosen threshold values and that for all threshold values the region where good odometer performance is balanced with computational efficiency is relatively small. We conclude that if odometer performance has to be weighed against computational cost, there is relatively little room for trade, and that a robust optimum can quickly and easily be found.

Keywords: Robot vision, robot navigation, real-time systems, robust estimation

1. INTRODUCTION

A visual odometer estimates motion from a sequence of images, for example from a camera that is mounted on a mobile robot. If the camera views a scene that is sufficiently planar, camera motion can be estimated by detecting and describing well localized keypoints—also called salient points or interest points—in one image, matching them against the keypoints of a second image, and computing the homography that projects keypoints from the first image coordinate frame to the second image coordinate frame. Such a homography can be decomposed into rotation of the camera around its axes, translation of the camera along the x and the y axis of the camera plane, and translation along the z axis up to a scale parameter that corresponds to the distance between the camera and the plane in the scene from which the keypoints were taken. If this distance can be estimated (e.g., using sonar or the spread of laser pointers) the full three-dimensional motion of camera and robot can be computed. See Caballero et al. (2009) for an overview of recent methodology.

In principle a full homography can be computed from only four keypoints, provided that they are not co-linear. However, with only four keypoints it is impossible to verify that all of them were correctly matched, and any noise and imprecision in the localization of keypoints will be left uncorrected. If more than four keypoints are available, least square methods for calculating the homography can reduce the effect of noise and imprecision (Ma et al., 2004). For a discussion on the covariance of projective homographies (Negahdaripour et al., 2005). With redundant pairs of keypoints, pairs that are statistical outliers can be discarded as likely mismatches, which further improves the reliability of the estimate. In practice we find that a homography that is computed from less than six pairs of keypoints cannot be trusted, and that even a homography that is computed from seven seemingly consistently placed pairs of keypoints occasionally turns out to be corrupted by false positives from the matching process. This raises the practical question of how many keypoints should be extracted in order to guarantee a reliable motion estimate.

Keypoints can be defined from a number of image features like corners, straight lines, or blobs, i.e., regions that are brighter or darker than their surroundings. While corners and lines are dominant in human made environments, blobs are dominant in natural environments. Here we consider the specific case where the camera of an autonomous submarine is trained on the sea floor, and where the dominant image features are patches of vegetation, rocks, shells, or worm holes, all of which are best described as blobs. Blob detectors typically define a blob by four principal parameters: its x and y coordinates in the image, the diameter or scale of the region that maximizes the contrast between image points inside the region and image points immediately outside of the region, and the difference in contrast between the blob and its surrounding,

[1] This research was supported by the European Commission's Seventh Framework Programme FP7/2007-2013 under grant agreement 248497 (TRIDENT Project)

which defines the blob response. Blob detectors and their implementations differ in the shape of regions that can be a blob, the radius of the region outside the blob that is taken into consideration, and their dependence on the contrast gradient along the edge of the blob.

In order to match blobs under rotation and change of scale, they need to be scale and rotation invariant. While the theory of scale and rotation invariant keypoints was developed during the 1980s (Witkin, 1983; Babaud et al., 1986; Lindeberg, 1990), the evolution of practical solutions for real time visual odometry took another two decades. We consider the SURF algorithm (Bay et al., 2008) and its various derivatives to be the first practical solutions. SURF uses integral images to generate the scale space and the Hessian determinant to identify a blob in an image, which makes it both significantly faster and more accurate than previous solutions like SIFT (Lowe, 2004), which is based on Gaussian integration and the less precisely localized Laplacian.

Implementations of visual odometers that track keypoints can be divided into a detection phase that identifies an often large set of keypoints with a maximal response in the local neighbourhood, a selection phase where the set of keypoints is reduced to the desired size and quality, a description phase where the selected keypoints are provided with a meaningful binary descriptor, a matching phase where descriptors of keypoints in different images are compared, and lastly the computation of the homography, which typically includes some method to separate the false positives of the matching phase from true positives. The algorithmic complexity of the detection and selection phases are generally of order $\mathcal{O}(pk)$ where p is the number of image pixels and k the number of considered scales, i.e., the number of considered blob sizes. For most applications p and k can be considered constant. The algorithmic complexity of the description phase is $\mathcal{O}(n)$, with n the number of selected keypoints. The worst case algorithmic complexity of the matching phase and of the computation of the homography is $\mathcal{O}(n^2)$, though we find that this can be improved upon easily by a variety of search heuristics. Speaking in more practical terms we find that at 100 keypoints per image and working at the relatively low resolution of 288 by 384 pixels, roughly 40% of the computation time of an unmodified open source SURF implementation goes into keypoint detection and selection, 40% into keypoint description, and about 20% into matching and calculation of the homography. For SIFT we measure a 10-fold increase in detection and description cost, skewing the previously described distribution of computational resources to 49%, 49% and 2%.

If keypoints are selected by a response threshold, as many authors of feature detection algorithms propose, the number of keypoints varies greatly between images, and with that the overall computation time. For example, in our experiments a threshold parameter of 500 (a value taken from OpenCV example code) in the OpenCV implementation of SURF results in an average of 190 keypoints that pass the threshold per image, out of an average of 570 extrema per image. Individual images however have anywhere between 1 and 600 keypoints that pass the threshold, see Figures 1. Accordingly, the computational cost differs by orders of magnitude between images.

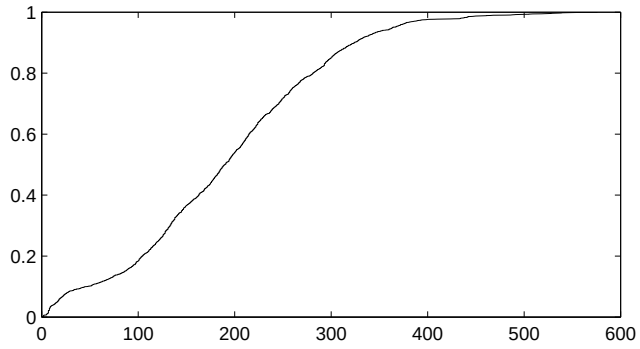

Fig. 1. Cumulative density function of the number of keypoints per image that pass a specific threshold response value. The x axis shows the number n of keypoints, the y axis the cumulative probability $P(\leq n)$ that an image has n or less keypoints that pass a specific threshold response. The statistic is based on 1,000 image pairs.

For real time systems it is preferable to select a fixed number n of keypoints per image. Keypoints can be sorted by response, and only the n keypoints with the highest response are selected. In this way the average and minimum response value of the extracted keypoints can vary greatly between images. The natural question that arises from such an approach is whether n can be freely set to satisfy constraints on the computational resources and the quality of the motion estimates, or whether n has narrow bounds, i.e., a lower bound below which no useful odometry is possible, immediately followed by an upper bound above which no further improvement to the motion estimate can be measured.

The remainder of this article describes our empirical approach to answer this question. Section 2 contains the experimental setup for the image acquisition and the organization of image pairs to be studied. Section 3 explains the statistical analysis of the error of the motion estimate as a function of n. Section 4 concludes.

2. EXPERIMENTAL SETUP

We base our analysis on two image sequences of more than 1,000 images each that were collected by a downward looking camera aboard the Girona500 autonomous submarine during two controlled test dives in the laboratory pool of the VICOROB research group in Girona. Image resolution is 384 by 288 pixels, and images are recorded at 3 frames per second. Lighting conditions were good and the water was clean, such that motion blur and light scatter are negligible. The floor of the pool was covered by a life-size high resolution color poster that shows a coral reef off the coast of Florida. The poster was spread out flat. All image features are co-planar. These are perfect conditions for the estimation of homographies. Any error in the motion estimates must be almost entirely attributed to the visual odometer and its configuration.

By matching the collected images against the digital version of the poster, a reasonable ground truth can be established. That is, the homography between each

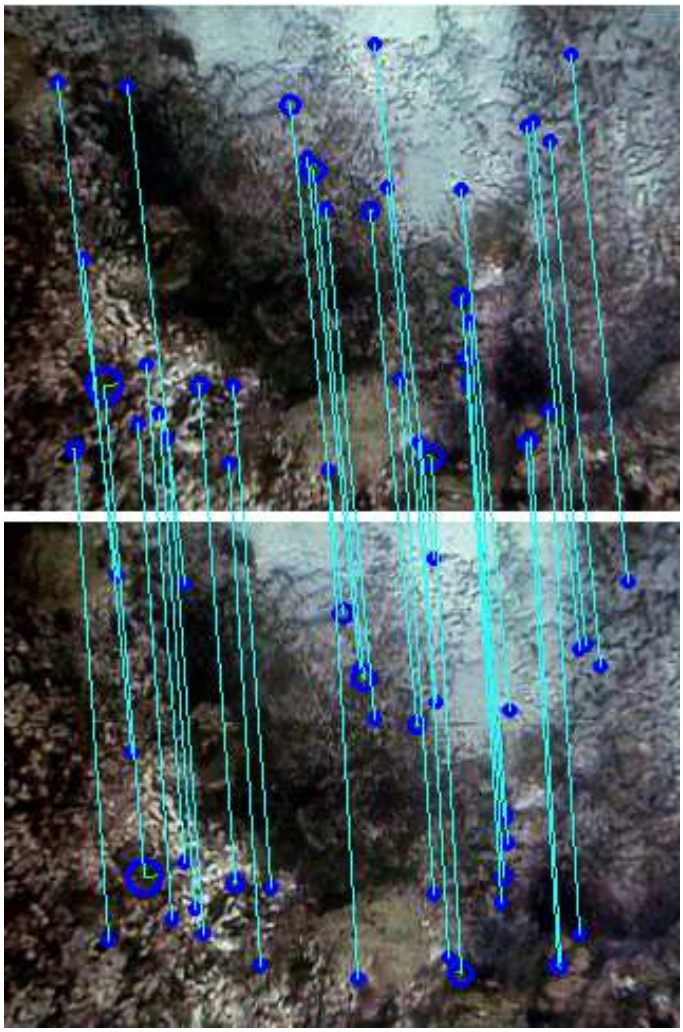

Fig. 2. Matching keypoints between consecutive images

image and the poster is computed from a large number of matching keypoints—typically 60 or more—that are distributed over the entire image, while individual pairs of images have smaller numbers of matching keypoints that are distributed only over the overlapping part of the images.

The back-projection error is the average Euclidean distance between the projection of a keypoint from the first image onto the second image, and the actual location of the matching keypoint in that second image. By evaluating the back-projection error on pairs of matching points that were not involved in the computation of the homography—i.e., by using cross-validation—we find that the back-projection error of the concatenation of a homography from an image to the poster and from the poster to another image is 0.5 pixels, with little deviation. That is, with the methods applied here 0.5 pixels is the minimal imprecision in the localization of a keypoint between images. Figure 2 shows a pair of images with matching features. The robot path over the poster is shown in Figure 3.

The robot moved at a constant altitude of one meter above the floor of the pool. Each of the two dive paths consists of ten stretches of either two or four meters length with

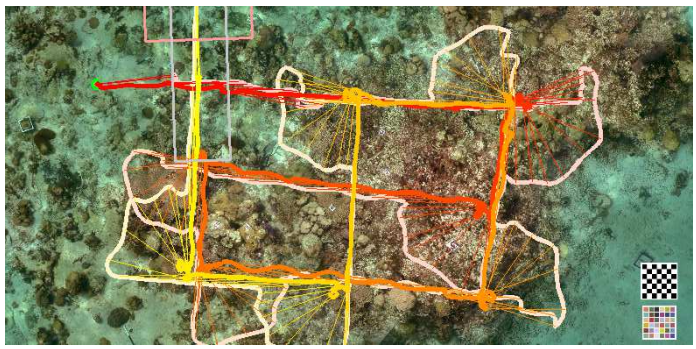

Fig. 3. Robot path over the poster, as estimated from the homographies. The bright line shows the camera movement, the saturated the robot center.

nine right angle turns in between, five of them to the right and four to the left. During the straight runs, the robot occasionally slowed down, rotated to adjust its path, and continued. Roll and pitch of the robot were negligible. Though we have calculated full homographies, we cannot report any observation on roll and pitch that exceeds the observed measurement error. In fact, the corresponding lower row matrix entries of the homographies are so close to zero that the homographies can be considered to be affine. In the remainder of this article yaw will be the only rotation considered in the analysis. All rotation values are measured in radians.

We use two independent open source implementations of SURF, one by C. Evans (Evans, 2009), and one included in the OpenCV image processing library that is being developed and distributed by Willow Garage. We used both implementations without modification. For reference we also report results on SIFT, again as included in the OpenCV library by Willow Garage. In all three cases feature descriptors are vectors of 64 float values that describe the region around the keypoint.

For the purpose of feature matching we have to distinguish between similarity and identity. The Euclidean distance between descriptors of features of the same type—i.e., worm holes on a background of sand or patches of sea grass on rock—is typically smaller than what can be expected for random features, and larger than for descriptors of the same feature in different images. The typical distance for similar and identical features is not universal but depends on the actual type of feature and the image quality. For example, the median Euclidean distance between descriptors of the same worm hole feature in different images might be 0.2 and the median distance to other worm hole features might be 0.3, whereas the median Euclidean distance between the same sea grass feature in different images might be 0.3 and the median distance to other sea grass features might be 0.5. A global threshold on the Euclidean distance will not be able to distinguish between a identical sea grass features and worm hole features that are merely similar.

To overcome this difficulty, we calculate the Euclidean distance of every keypoint $a \in A$ in one image to every keypoint $b \in B$ in another image and record the keypoint b' with the smallest distance to a and the keypoint b'' with the second smallest distance to a. A match between a and

b' is only confirmed if the distance between a and b' is less than two third of the distance between a and b''. This effectively creates a threshold that dynamically adjusts to the image and descriptor quality and works well as long as the number of keypoints that are matched against each other for a pair of images is not too small, e.g., not smaller than 10.

For the computation of the homography we use RANSAC (Random Sample Consensus, Fischler and Bolles, 1981) in combination with least squares. RANSAC is used with a threshold value of two, i.e., matching pairs are considered to be outliers if the back-projection error of the homography exceeds two pixels. We will use the term "inliers" to indicate matching keypoint pairs that were considered inliers by RANSAC.

We considered pairs of images that are consecutive (distance one) and that are of distance $\{2, 3, \ldots, 29, 30\}$ in the two image sequences. This makes for over 60,000 image pairs. Average motion between images is ten pixels, mostly along the vertical axis, such that two consecutive images overlap by 96%, and images at distance ten still overlap by two third of their area. Images at distance thirty only overlap if the robot slowed down or turned. For each image pair we selected the same number n of keypoints based on highest response, i.e., based on the Hessian determinant in the case of SURF, and the Laplacian in the case of SIFT. For each value of n we compare all 60,000 image pairs. We use values of $n \in \{10, 20, \ldots, 490, 500\}$ keypoints, i.e., n grows in increments of ten.

If less than four pairs of keypoints can be matched between images, no homography can be calculated and no motion estimate can be given. We also ignore homographies where RANSAC could find only four or five inlier pairs as too unreliable without further refinement. Statistical results for such low numbers of inliers depend largely on the further application of filter heuristics, which exceeds the scope of this article. For example, such estimates must be rejected if the corresponding keypoints form an almost colinear set or if the descriptors of the matching pairs are dissimilar enough to allow for false positives, a case which poses no problem in larger sets of matching pairs.

3. ANALYSIS

To quantify the quality of a motion estimate we distinguish between the error in spatial translation, error in rotation about the z axis (yaw), and error in the scale. We define the translation error $e_{x,y}$ of an image pair to be the Euclidean distance $\sqrt{(x-x')^2 + (y-y')^2}$ between the translation vector (x, y) of the motion estimate that is based on the homography between images, and the translation vector (x', y') as estimated by way of the poster. We define the rotation error e_{yaw} of an image pair to be the absolute difference between yaw as measured from the homographies between images, and yaw as measured by way of the poster. From a homography the robot motion along the z axis of the camera can be computed only up to scale, which corresponds to the distance between camera and scene. Since the precision in the estimation of this distance does not concern us here, and since scale varies only slightly, always being close to one, we will only consider an error in the estimation of its value. We define the scaling error e_{scale} of an image pair to be the absolute difference between the relative scale as measured from the homography between two images and the relative scale as measured by way of the poster.

As the number n of keypoints that are extracted from an image and matched against another image increases, the number of correctly matched pairs of keypoints generally increases as well, and the errors of the corresponding motion estimates decrease. For some image pairs however the number of matching keypoint pairs is either too low to compute a homography, or the number of inliers selected by RANSAC is too low for the estimate to be reliable. We have observed cases where the mean translation error decreases when less keypoints are extracted, while the number of image pairs for which no reliable motion estimate can be calculated increases. If we wish to quantify the quality of the motion estimate that can be expected for a given number of extracted keypoints, both the error of the estimate and the number of images for which motion cannot be reliably estimated need to be taken into account.

To quantify the quality of a motion estimate as a function of extracted keypoints per image we count the number of image pairs that pass certain quality thresholds. The lowest quality threshold, threshold I, counts the number of image pairs with at least six inliers as selected by RANSAC. Three more thresholds count the number of image pairs with translation, rotation and scaling errors that are all lower than some given error levels. To choose suitable values for these error levels consider Figure 4, which shows the Euclidean translation error $e_{x,y}$, the absolute rotation error e_{yaw} and the absolute scaling error e_{scale} as a function of the number of pairs of matching keypoints from which motion was estimated. The number n of extracted keypoints was hundred for all images. For this statistic we considered image pairs that were consecutive (distance one) and that were of distance $\{2, 3, \ldots, 29, 30\}$ in the two image sequences, for a total of over 60,000 image pairs.

Figure 4 shows that for all three types of error and for the three considered detectors the graphs can be divided into three regimes. Under the first regime the errors decrease rapidly until the sets of matching keypoints reach size 20. Under the second regime further but less dramatic decrease can be observed until the sets of matching keypoints reach size 40. From then on, in the last regime, there is little to no further decrease. Absolute error values are nearly identical for the two implementations of SURF, and quite similar even for SIFT. Note that the two SURF implementations have larger maximum match sizes than SIFT only because they have a larger likelihood of false positives in the matching process. The average number of true positives obtained by SURF is actually smaller.

We consider the errors at matching sets of size 12 to define threshold II, the errors at matching sets of size 20 to define threshold III and the errors at matching sets of size 40 to define threshold IV. Each of the thee thresholds consists of three error levels $e_{x,y}$, e_{yaw}, and e_{scale}, where each error level is the highest error observed for any of the three detectors for the respective number of matching sets, which is always the error of the SIFT detector. The definitions of the three thresholds are given

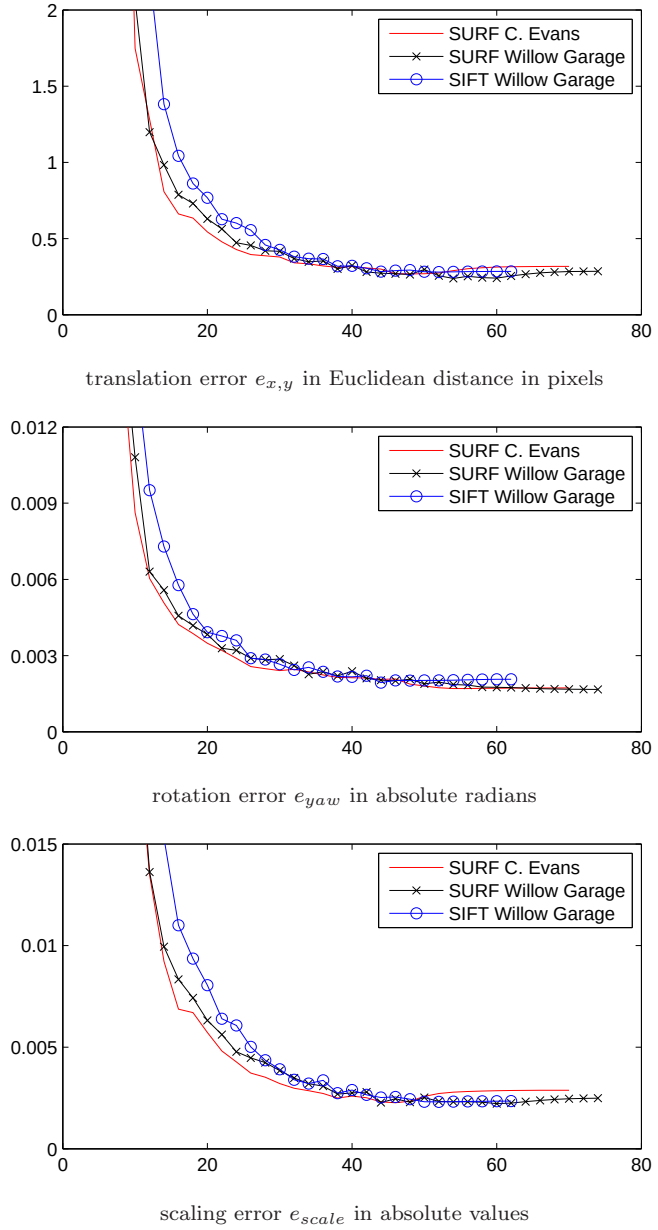

Fig. 4. Magnitude of different error types as a function of keypoint pairs on which the motion estimate was based. The x axis shows the number of matched pairs. The y axis shows the average error.

in Table 1. Note how error levels approximately double between threshold IV and III, and again between threshold III and II.

Figure 5 reports the percentage of motion estimates that pass every threshold as a function of the number of keypoints that are extracted per image. Percentage values for selected numbers of keypoints per image are given in Table 2. Except for the obvious fact that for each threshold the graphs converge on different values, the graphs for different detectors and for different thresholds are very similar one to the other. For all implementations the percentage of image pairs that pass all thresholds increases dramatically until about $n = 50$ keypoints are extracted per image. There is a period of transition until about

Table 1. Definition of quality thresholds for motion estimates.

	at least 6 inliers		
threshold I			
threshold II	$e_{x,y} \leq 1.82$	$e_{yaw} \leq 0.0080$	$e_{scale} \leq 0.0211$
threshold III	$e_{x,y} \leq 0.75$	$e_{yaw} \leq 0.0041$	$e_{scale} \leq 0.0079$
threshold IV	$e_{x,y} \leq 0.33$	$e_{yaw} \leq 0.0023$	$e_{scale} \leq 0.0029$

Table 2. Percentage of motion estimates that pass a threshold for selected numbers of extracted keypoints per image.

	number of extracted keypoints						
detector	20	30	50	100	200	300	500
	threshold I (at least 6 inliers)						
SURF C. Evans	9.0	18.7	31.0	44.7	54.6	59.0	62.3
SURF W. G.	13.6	24.2	35.8	47.9	57.0	61.1	64.8
SIFT W. G.	6.7	17.4	33.8	52.4	64.9	70.3	75.4
	threshold II						
SURF C. Evans	14.1	21.3	30.1	39.7	46.5	49.3	51.4
SURF W. G.	19.0	26.2	34.0	42.2	48.3	50.9	53.2
SIFT W. G.	8.6	16.2	28.2	41.8	50.7	54.1	57.4
	threshold III						
SURF C. Evans	8.4	13.8	20.8	28.1	32.4	34.4	35.7
SURF W. G.	12.1	17.9	24.1	29.9	33.5	35.1	36.5
SIFT W. G.	5.0	9.8	18.7	28.8	34.8	36.7	38.3
	threshold IV						
SURF C. Evans	3.0	5.3	9.1	12.7	15.4	16.3	16.9
SURF W. G.	5.2	8.2	11.3	14.4	16.2	16.9	17.5
SIFT W. G.	1.7	3.7	8.3	13.6	16.5	17.4	18.0

$n = 200$ during which the increase continuous on a clearly measurable scale for all thresholds. Increasing n further shows little results, in particular for the stricter thresholds III and IV.

4. CONCLUSION

We first addressed the methodological problem of how to measure odometer performance when no reliable motion estimate can be given for a large number of image pairs. We concluded that the quality of motion estimates for a given number of extracted keypoints per image can best be quantified by the percentage of image pairs that pass given quality thresholds. The exact values of the quality thresholds determine to which value the percentage of image pairs will converge when more and more image keypoints are selected, but they do not affect the general shape of the graph. This is of practical use because it allows the practitioner to produce a cost-benefit analysis for the number of selected keypoints that is relatively invariant to quality thresholds.

With regard to our main research question, whether the number of extracted keypoints can be set freely to satisfy quality and resource constraints, the answer is negative. There is some room to trade performance against resource consumption, but the range of values for which this is possible is narrow and well defined. In this experiment, lowering the number of keypoints below 50 leads to a rapid deterioration in performance, while raising it above 200 promises very little improvement.

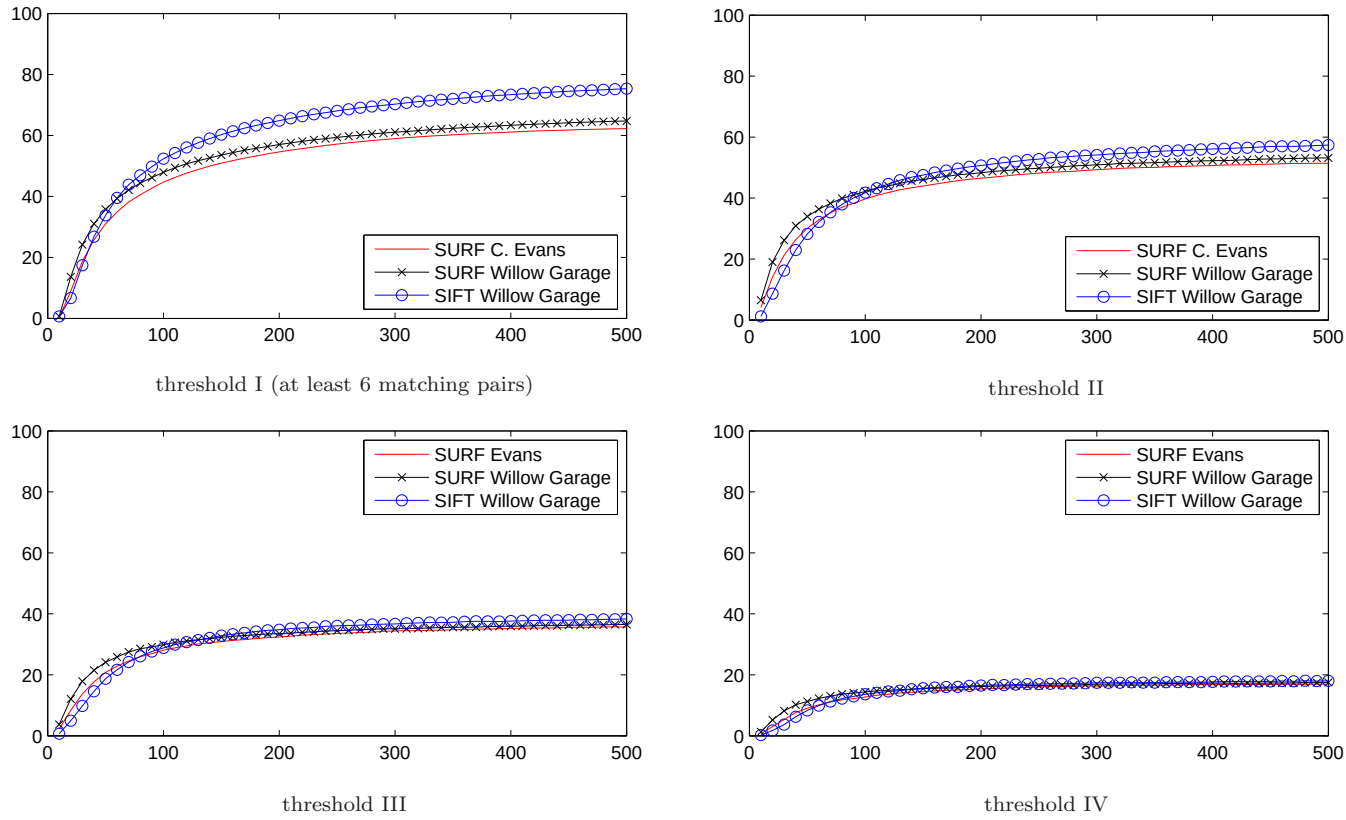

Fig. 5. Percentage of motion estimates that pass a given quality threshold as a function of extracted keypoints per image. The x axis shows the number of keypoints extracted for each image in a pair. The y axis shows the percentage of image pairs that pass the threshold.

Our overall experience from the experiments described here is that it is impractical to fine-tune the exact number of extracted keypoints to the actual constraints of speed and accuracy, average keypoint quality, or average amount of overlap between images. A setting of 100 keypoints per image quickly emerged as the practical optimum.

ACKNOWLEDGEMENTS

We thank Joan Pau Beltran, Nuno Gracias, and the Computer Vision and Robotics (VICOROB) group in Girona for valuable comments and support.

REFERENCES

Babaud, J., Witkin, A.P., Baudin, M., and Duda, R.O. (1986). Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1), 26–33. doi:10.1109/TPAMI.1986.4767749.

Bay, H., Ess, A., Tuytelaars, T., and van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359. doi:10.1016/j.cviu.2007.09.014.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2009). Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *Journal of Intelligent and Robotic Systems*, 54(1-3), 137–161. doi:10.1007/s10846-008-9257-y.

Evans, C. (2009). Notes on the OpenSURF Library. Technical Report CSTR-09-001, University of Bristol.

Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. doi:10.1145/358669.358692.

Lindeberg, T. (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3), 234–254. doi:10.1109/34.49051.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Ma, Y., Soatto, S., Kosecka, J., Ma, Y., Soatta, S., Kosecka, J., and Sastry, S. (2004). *An invitation to 3-D vision*. Springer, Berlin / Heidelberg.

Negahdaripour, S., Prados, R., and Garcia, R. (2005). Planar homography: accuracy analysis and applications. In *Image Processing, ICIP'05. IEEE International Conference on*, I–1089–92. doi:10.1109/ICIP.2005.1529944.

Witkin, A.P. (1983). Scale-space filtering. In *Artificial Intelligence, IJCAI'83. Proceedings of the International Joint Conference on*, 1019–1022.