

Affine Invariant Feature Tracker Evaluation based on Ground-Truth Data

Zoltán Pusztai^{1,2} and Levente Hajder^{1,2}

¹ Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary

² Department of Algorithms and Their Applications, ELTE IK, Budapest, Hungary

Abstract

Affine invariant feature detectors attract more and more attention these days. They can retrieve more information from the images than regular detectors that can detect feature point correspondences only. Our paper determines the reliability and accuracy of the detectors and quantitatively compares the affine transformations provided by them. The comparison is based on ground-truth data. They are generated by scanning real-world objects using a high-resolution structured-light 3D scanner.

1. Introduction

Feature point detection and tracking is a well-studied problem in the field of computer vision. Many algorithms rely on the correctly detected features and their matches between images like the ones used in panorama matching, motion detection, focal length estimation⁵ or Simultaneous Localization and Mapping¹⁶. There are well-known detectors in the field, without the need for completeness, e.g. AKAZE¹⁰, BRISK⁶, GFTT²⁰ (Good Features To Track – also known as Shi-Tomasi corners) and SIFT⁷.

However, the above mentioned detectors do not provide affine transformations, only the plain locations. The affine transformations are 2D transformations between the infinitively small area of the feature points. They are computed from the affine regions, which are provided by the affine invariant detectors. These transformations receive more and more attention from the research community nowadays. If the transformations are known, it is possible to increase the accuracy of the algorithms or measure properties of the camera movement from less number of correspondences¹⁵.

Many comparisons have been done between the different feature detectors and trackers⁴. Maybe the best known is the Middlebury database[†] that consists of several datasets and have been continuously developed since 2002. In the first period, they generated corresponding feature points of real-

world objects¹⁸ which is used for the comparison of feature matchers. Later on, this stereo database was extended by novel datasets using structured-light¹⁹ conditional random fields¹¹. Even sub-pixel accuracy can be achieved in this way as discussed in¹⁷.

The authors of this paper also published a study that evaluates the feature location errors of the detectors implemented in OpenCV3¹². One of our other paper¹⁴ deals with the affine invariant feature detectors but only the location error has been compared in that paper, the affine invariant transformation was used only for matching the feature points between the images.

This paper is the extension of the latter comparison. Instead of comparing the feature locations, we wish to compare the affine transformations between the matched image features, based on ground-truth (GT) data. To the best of our knowledge, this kind of transformation evaluations has only been done in the paper published by Mikolajczyk et al.⁹. Their extensive comparison includes viewpoint and rotation change, zooming, image blurring, JPEG compression and light change. However, all of the images were captured about static scenes for which the objects do not move and self-occlusion is not present. The goal is to compare the feature detectors in a more real-world like scenarios: the objects are ragged and features appear and disappear due to self-occlusion.

The main contribution of this paper is the evaluation of the accuracy of affine invariant feature detectors based on

[†] <http://vision.middlebury.edu/>

ground-truth (GT) data. The data are generated using our high precision structured-light scanner and the detectors are tested mainly against rotation movement. The following novel feature detectors are compared: Harris-Laplace, Hessian-Laplace, Harris-Affine⁸, Hessian-Affine⁸, IBR²¹, EBR²², Edge-Laplace and SURF³.

2. Ground Truth Data Generation

GT data are needed for the quantitative evaluation of the above mentioned, affine-invariant feature detectors. A high precision structured-light 3D scanner is used for the data generation.

2.1. 3D Scanner Calibration

The precision of the scanner is based mainly on the accuracy of the calibration of its components. The 3D scanner consist of three components: a camera, a projector and a turntable. The calibration of the camera is carried out by the well-known method of Zhang²³. The projector can be described by an inverse camera model, thus the same method can be used for the projector calibration as well. In our approach, a gray-coded structured-light encodes the projector pixels by black and white stripped patterns. The locations of the chessboard corners in projector space can be acquired by decoding the structured light from the camera images individually. The last component of the scanner, the turntable, is calibrated using the chessboards as well. This time the centerline of the turntable needs to be precisely determined, thus the chessboard is rotated, elevated, and its corners are reconstructed in 3D. The detailed calibration of the scanner can be found in our recent paper¹³.

2.2. 3D reconstruction of test objects

After the calibration is finished, the structured-light scanner can be used for 3D scanning. The test objects are placed on the turntable one by one, and structured light is projected on them. Since the scanner is already calibrated, the 3D coordinates of the object can be calculated by triangulation⁵. We used 3 degrees rotations between the different scans, and altogether 20 scans of each test object, which means they are rotated by 60 degrees. Seven different real-world object are used in the comparison. These are the following:

1. Bag(12.4M): Darkish, ragged surface.
2. Books(14.4M): Two good textured books placed on each other.
3. Cube(14.3M): Huge object with flat sides, the textures contain large homogeneous regions.
4. Flacon(7M): Big object with mostly homogeneous texture.
5. PlushDog(3.6M): Medium-sized low-textured plush toy.
6. Poster(6.4M): Well-textured planar object, this is an easy test case for the detectors.

7. T-Rex(8.8M): Small plastic toy with homogeneous dark texture.

The numbers in the parentheses are the amount of spatial points reconstructed by the scans. See the top row of Fig. 1 for the reconstructed point clouds. The bottom row shows example images of each test object.

The test objects are selected to cover many challenging situation for the trackers. On the small objects, the detector cannot find big features with large shapes which are mostly easier to follow. The big objects provide self-occlusion when rotating, thus features are appearing and disappearing in the sequence. Low-textured objects are also used to make the task challenging, however an easier test object (Poster) is also represented. The latter one provides no self-occlusion, it has highly textured regions, its motion is only a rotation.

2.3. GT affine transformation generation

Since the point cloud of the test objects and the parameters, both intrinsics and extrinsics, are known, it is possible to calculate the GT affine transformations between the image features. The calculation in our approach is as follows:

First, the tangent plane is calculated for each spatial point within the point cloud. For this task, the spherical neighborhood of the points is taken into account with radius 0.1 cm. Note that the point cloud usually contains 4-14 million points, for the individual number of points for the test cases see the enumeration located in Sec 2.2, thus the tangent plane can be precisely computed by Principal Component Analysis (PCA).

Then 4 points of the tangent plane has been selected and re-projected into the camera image using the calibrated camera parameters. The points in the tangent plane and the feature point itself are then rotated around the centerline of the turntable[‡] and re-projected again to the next camera image. The re-projection of the rotated 3D feature point gives the GT location of the feature in the next image. A homography can also be calculated using the tangent plane and the camera parameters. The affine transformation between the image features on the subsequent images can be calculated from this homography². Let \mathbf{H} be a homography between the re-projected points of the two tangent planes:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \quad (1)$$

and let $\mathbf{u} = [u_x, u_y]^T$ and $\mathbf{v} = [v_x, v_y]^T$ be the GT feature locations in the succeeding images, and $s =$

[‡] The turntables is also calibrated, therefore its axis is known. The rotation angle can be precisely retrieved from the stepper motor.

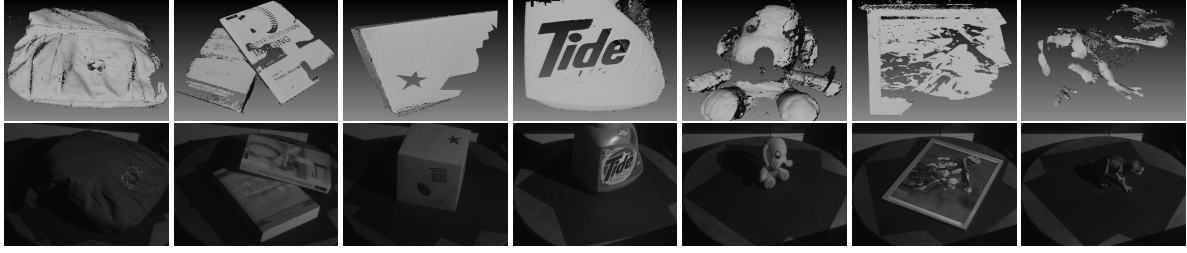


Figure 1: Objects used for testing in columns: Bag, Books, Cube, Flacon, PlushDog, Poster and T-Rex. Upper row: the reconstructed point clouds. Bottom row: first images of the corresponding sequence.

$[h_{31}, h_{31}, h_{33}][u_x, u_y, 1]^T$. Then the elements of the A affine transformation are given as

$$\begin{aligned} a_{11} &= \frac{h_{11} - h_{31}v_x}{s}, \\ a_{12} &= \frac{h_{12} - h_{32}v_x}{s}, \\ a_{21} &= \frac{h_{21} - h_{31}v_y}{s}, \\ a_{22} &= \frac{h_{22} - h_{32}v_y}{s}. \end{aligned} \quad (2)$$

The dimension of the A matrix is 2 by 2, because the translation part is already known by the feature points location ($\mathbf{t} = \mathbf{u} - \mathbf{v}$).

3. Recover Affine Transformation from Affine Regions

The implementations of the mentioned feature detectors are downloaded from the website of University of Oxford[§]. These feature tracking algorithms usually work as follows: the detector separately finds the features location and local affine transformation in the images. Then a feature descriptor is applied to describe the area of the features as a vector and the features between the images are matched based on the norm between these feature vectors. The final affine transformation between the \mathbf{u} and \mathbf{v} image features – lying on different images – can be calculated from their local affine transformation:

$$\mathbf{A} = \mathbf{A}_2^{-1} \mathbf{A}_1, \quad (3)$$

where \mathbf{A}_1 and \mathbf{A}_2 are the local affine transformations at locations \mathbf{u} and \mathbf{v} , respectively.

However, the implementations do not directly give the local affine transformations, they only specify the affine regions of the features. Let us denote the three parameters of the regions as a , b and c . These parameters are the elements of the second-moment matrix (\mathbf{M}) of the related affine regions, described as

$$\mathbf{M} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}. \quad (4)$$

The affine regions of the ellipses can be described as an implicit form: $a(x-x_0)^2 + 2b(x-x_0)(y-y_0) + c(y-y_0)^2 = 1$, where $[x_0 \ y_0]^T$ and a, b, c are the location of the image feature, and the parameters of the related second-moment matrix. The directions and lengths of the axes can be calculated considering the eigenvectors and eigenvalues of \mathbf{M} . The square root of this matrix is a transformation which normalizes the region of the ellipse into that of a circle.

The eigenvectors can be zero. The ellipse does not change if we multiply its eigenvectors by -1 , thus there is an ambiguity to the axes. We restrict the axes to form a left handed coordinate system, and eliminate the ambiguity by normalized cross correlation between the image patches.

However, we aim to recover the affine transformation between the image features and unfortunately there is an unknown rotation between the normalized image patches, but it can be recovered using geometrical properties. We decided to rotate the major axis of the ellipse to match with the x axis on both patches. Let us denote these rotations by matrices \mathbf{R}_1 , \mathbf{R}_2 for the first and second patches, respectively. Then the affine transformation between the image features is given by:

$$\mathbf{A} = \mathbf{M}_2^{-\frac{1}{2}} \mathbf{R}_2^{-1} \mathbf{R}_1 \mathbf{M}_1^{\frac{1}{2}}, \quad (5)$$

where \mathbf{M}_1 , \mathbf{M}_2 are the second-moment matrices for the first and second affine patches. See Figure 2 for better understanding.

4. Comparison

After the original and the GT transformation is computed, we can compare them. The comparison is based on the Frobenius norm of the difference of the matrices, which has a geometrical meaning: as Barath et al. showed¹, the L_2 norm of the affine transformation is equivalent on the norm of the vectors effected by the transformation.

Two comparisons are carried out to test the affine invariant

[§] <http://www.robots.ox.ac.uk/vgg/research/affine/>

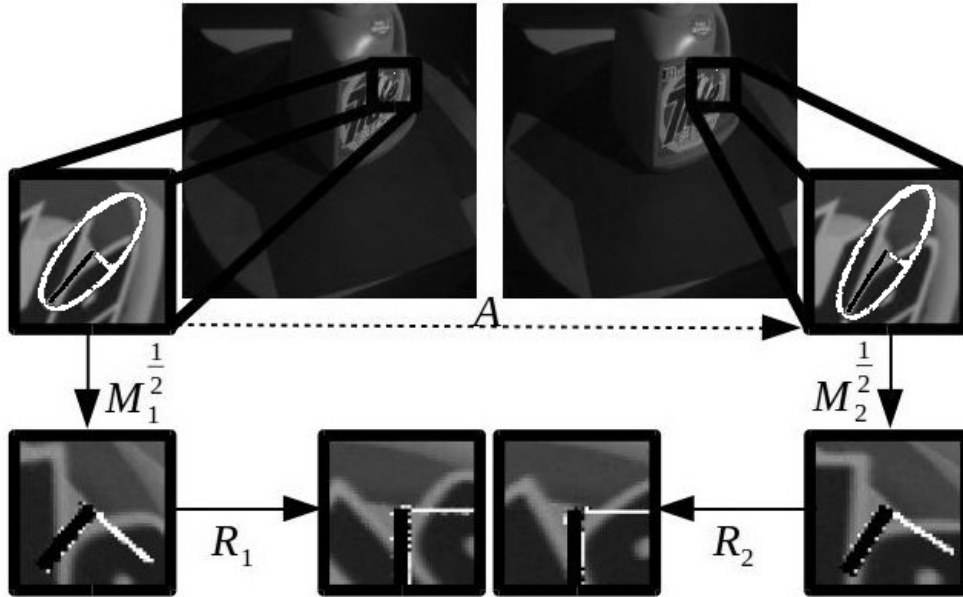


Figure 2: Affine transformation (**A**) retrieval. The second-moment matrices (M_1 , M_2) are used to normalize the affine patches. The unknown rotation between the two patches is retrieved by rotating the ellipses.

feature detectors. We propose two comparisons: (i) the first one follows the traditional way of feature matching, (ii) the second one aims to measure the pure differences between the detectors.

4.1. Traditional Matching

This comparison follows the usual way of feature matching between the images. First, the features are separately detected in the images, then a feature descriptor is used to build a descriptor vector, then these vectors are compared based on their Euclidean distance, finally the nearest ones form a match of features between the images. In the test, the popular SIFT descriptors defined in ⁷ is applied, and the ratio test, proposed by D.Lowe in the same paper ⁷, is used for matching due to its robustness.

Table 1 shows the result of the matching. The columns are the tested methods, each row is a different test object. The sub rows in each test case are as follows:

- 'All': the average of all feature points found in all the 20 images of the testing sequences.
- 'Matched': the average number of matched points.
- 'Cant Rec.': the average number of matched features, which could not be reconstructed.
- 'Det. Err': the average number of detections errors.
- 'Aff. Err': the average number of affine errors.

The description of this values are as follows: The rows 'Cant Rec.' show the number of features that fell off the object or cannot be reconstructed due to some effect of

the structured-light scanner such as self-occlusion. The GT location of each feature can be calculated by reconstructing, rotating around the centerline of the turntable, and re-projecting it to the next image. A detection error ('Det. Err') is found when the location of the matched feature is at least 3.0 pixels away from its GT location. In this case it makes no sense to compare the related affine transformation with the GT, because the processed feature is not found correctly. The last sub-rows ('Aff. Err.') shows the average number of features that are accurate enough for evaluating the affine transformation.

Table 2 shows the average and median errors of the matchers. It can be seen that despite the huge number of feature found by the Harris-based detectors, only 30% – 40% of features are matched. Other detectors like SURF, EBR or IBR found significantly less number of features, however the matching rate is over 50%.

Despite the fact that SURF finds the least number of features on the test cases, it has the lowest affine error rate. The corner based method (EBR) and IBR usually perform better than the others in case of feature tracking, however their ability to estimate the affine shape is average. The Hessian methods (HESAFF, HESLAP) offer well estimated affine transformations and their ability to follow the features is the second one after EBR and IBR, that is they provide a good compromise.

4.2. Ground-truth Data-based Matching

Unfortunately, the quality of the 'traditional' matching is not convincing because of inaccuracy. Sometimes, more than 50% of the features are dropped after the matching due to the ratio criteria. Thus another test has been carried out: the feature matching is based on the GT locations of the features, matching algorithms are not applied. This is not the case in the real-world scenarios because of the lack of GT, however, this way the error of the false matching is eliminated and the pure error of the affine detectors can be measured.

The feature matching is as follows: the GT location of the feature on the next frame is calculated, as it is described in the previous subsection. Algorithmic feature matching is not applied here due to the inaccuracy. Then the nearest feature is selected as GT location. If the distance between the nearest and the GT is more than 3.0 pixels, then the feature match is calculated as a detection error, otherwise the affine error between the affine transformations is calculated.

Similarly to the first test, SEDGELAP algorithm finds the most number of feature on each test case. The Harris-based methods (HARAFF, HARHES, HARLAP) find 30 – 40% of SEDGELAP, and SURF finds the least number of features. See Table 3 for the details.

SEDGELAP has also the lowest detection errors in each test (see Table 4). That is not surprising, because the nearest feature is selected to the GT, and SEDGELAP has huge number of features on the test images. Thus, it is expected that it outperforms the others. SURF has the lowest affine error rate, followed by EBR and IBR, and the Hessian-based methods.

Remark that the median affine error is below the average in most of the cases. This means that the errors of the detectors are mostly low and there are just a few cases when error values are huge.

5. Conclusion and Future Work

In this paper, a method is presented for quantitative comparison of affine invariant feature detectors. Then GT data of real-world objects are generated by a 3D structured-light scanner. The affine transformation is reconstructed using the affine shape of the features, the matching is done by either the SIFT descriptor, or based on the GT location of the features.

The tests show that SURF performs extremely well in all situations, despite the fact that it finds the lowest number of features. In contrast of SURF, SEDGELAP finds the most number of features in the images, however the affine transformations provided by them are the least accurate ones compared to all tested detectors. The HESSIAN-based methods, and IBR + EBR, provide high quality affine transformations, however, the former ones find more features. Harris-based methods find 3 – 5 times more feature then the

Hessian-based, but their affine transformations are less accurate.

Tables 1 and 4 contain several interesting results, however, the detailed description is out of the scope of this paper.

Acknowledgements



The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).



Supported by the ÚNKP-17-3 New National Excellence Program of the Ministry of Human Capacities.

References

1. D. Barath, J. Matas, and L. Hajder. Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry. In *Proceedings of the British Machine Vision Conference*, 2016. 3
2. D. Barath, J. Molnár, and L. Hajder. Novel methods for estimating surface normals from affine transformations. In *Computer Vision, Imaging and Computer Graphics Theory and Applications VISIGRAPP, Revised Selected Papers*, pages 316–337, 2015. 2
3. H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 2
4. J. Chao, A. Al-Nuaimi, G. Schroth, and E. Steinbach. Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with JPEG-encoded query images. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Pula, Sardinia, Italy, Oct 2013. 1
5. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 1, 2
6. S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2548–2555, 2011. 1
7. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, ICCV '99*, pages 1150–1157, 1999. 1, 4
8. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *In Proc. ECCV*, pages 128–142, 2002. 2
9. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005. 1

10. A. B. Pablo Alcantarilla (Georgia Institute of Technology), Jesus Nuevo (TrueVision Solutions AU). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. [1](#)
11. C. J. Pal, J. J. Weinman, L. C. Tran, and D. Scharstein. On learning conditional random fields for stereo - exploring model structures and approximate inference. *International Journal of Computer Vision*, 99(3):319–337, 2012. [1](#)
12. Z. Pusztai and L. Hajder. Quantitative comparison of feature matchers implemented in opencv3. In *Proceedings of the 21st Computer Vision Winter Workshop. CVWW 2016*, pages 1–9. SDRV, Ljubljana, 2016. [1](#)
13. Z. Pusztai and L. Hajder. A turntable-based approach for ground truth tracking data generation. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, pages 500–511, 2016. [2](#)
14. Z. Pusztai and L. Hajder. Quantitative comparison of affine invariant feature matching. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 515–522, 2017. [1](#)
15. C. Raposo and J. P. Barreto. Theory and practice of structure-from-motion using affine correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5470–5478, 2016. [1](#)
16. D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95(1):74–85, 2011. [1](#)
17. D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nescic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition - 36th German Conference, GCPR*, pages 31–42, 2014. [1](#)
18. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002. [1](#)
19. D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 195–202, 2003. [1](#)
20. Tomasi, C. and Shi, J. Good Features to Track. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 593–600, 1994. [1](#)
21. T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *In Proc. BMVC*, pages 412–425, 2000. [2](#)
22. T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, Aug. 2004. [2](#)
23. Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, Nov. 2000. [2](#)

Table 1: Each column shows a method, each row is a test case. The meaning of the sub-rows: 'All': number of detected features, 'Matched': number of features matched by SIFT, 'Cant Rec.': number of features that we could not track, 'Det Err.': number of matched features that has more than 3 pixel errors, 'Aff. Err.': number of matched features that are matched perfectly.

		EBR	HARAFF	HARHES	HARLAP	HESAFF	HESLAP	IBR	SEDGELAP	SURF
Bag	# All	22.32	1149.26	1173.00	1167.58	25.63	26.16	129.21	1292.42	8.89
	# Matched	9.89	354.00	359.21	357.53	10.05	10.05	68.00	452.26	7.32
	# Cant Rec	1.89	114.26	117.53	117.37	6.32	6.32	22.21	205.26	1.58
	# Det. Err	3.89	29.05	29.79	28.95	1.00	1.00	11.79	38.26	0.42
	# Aff. Err	4.11	210.68	211.90	211.21	2.74	2.74	34.00	208.74	5.32
Books	# All	309.73	7839.11	8246.84	7992.05	466.79	468.47	730.84	8699.26	152.68
	# Matched	163.11	2359.68	2525.95	2435.32	181.63	180.58	435.37	2962.47	127.32
	# Cant Rec	107.32	1463.58	1563.26	1517.84	99.42	98.32	279.32	1766.47	72.95
	# Det. Err	18.42	158.79	182.789	164.84	28.47	28.42	28.89	323.95	2.74
	# Aff. Err	37.37	737.32	779.9	752.63	53.74	53.84	127.16	872.05	51.63
Cube	# All	89.42	804.90	830.32	811.42	50.16	50.16	32.21	1663.21	8.05
	# Matched	55.95	251.11	259.42	252.21	17.68	17.68	21.42	428.37	6.42
	# Cant Rec	48.63	152.68	155.68	154.53	5.16	5.16	9.47	252.21	4.16
	# Det. Err	2.32	29.37	34.63	29.26	11.47	11.47	3.21	94.00	0.95
	# Aff. Err	5.00	69.05	69.11	68.42	1.05	1.05	8.74	82.16	1.32
PlushDog	# All	34.42	1401.47	1604.84	1419.53	326.74	336.05	127.58	3612.26	39.26
	# Matched	19.32	480.32	565.16	485.42	152.58	157.11	74.32	1189.53	32.63
	# Cant Rec	6.32	253.26	299.53	257.11	90.89	94.95	33.00	673.32	16.84
	# Det. Err	10.37	53.53	74.47	53.89	33.26	33.53	21.16	203.26	4.89
	# Aff. Err	2.63	173.53	191.16	174.42	28.42	28.63	20.16	312.95	10.89
Poster	# All	346.16	5945.89	6915.32	5998.00	1522.26	1544.47	591.47	13820.60	306.63
	# Matched	191.84	2237.32	2647.79	2268.84	661.58	672.37	372.58	4576.63	269.90
	# Cant Rec	104.84	1007.89	1209.79	1030.53	301.84	309.79	199.90	2239.16	142.63
	# Det. Err	37.79	78.53	112.32	79.89	38.53	38.53	32.84	425.32	5.89
	# Aff. Err	49.21	1150.89	1325.68	1158.42	321.21	324.05	139.84	1912.16	121.37
Flacon	# All	546.84	3062.00	3980.58	3119.42	1300.53	1355.63	286.68	7429.58	283.42
	# Matched	322.11	1095.05	1476.00	1126.58	514.90	530.53	179.00	2567.79	241.90
	# Cant Rec	136.37	583.47	787.68	592.05	300.58	308.63	87.32	1254.47	136.79
	# Det. Err	72.47	69.05	88.58	71.79	28.79	29.63	26.26	395.47	7.74
	# Aff. Err	113.26	442.53	599.74	462.74	185.53	192.26	65.42	917.84	97.37
T-Rex	# All	9.53	958.79	1048.74	973.53	150.37	155.63	77.84	2377.21	33.53
	# Matched	4.79	267.79	306.63	277.95	52.95	57.89	34.32	717.63	24.63
	# Cant Rec	1.63	158.05	184.63	167.53	34.79	39.26	13.11	542.26	9.37
	# Det. Err	1.84	22.53	26.84	23.16	7.05	7.53	7.79	54.21	3.47
	# Aff. Err	1.32	87.21	95.16	87.26	11.11	11.11	13.42	121.16	11.79

Table 2: This table contains the detection and affine errors of the methods (columns) on a test object(rows). Each error is represented by its median ('Med.') and average ('Avg.'). Affine error is computed if the matched feature is in the 3 pixel radius of the GT, otherwise detection error is computed. Det. Err. and Aff. Err are given by the reprojection error and the Frobenius norm of difference matrix of retrieved and GT affine matrices, respectively.

			EBR	HARAFF	HARHES	HARLAP	HESAFF	HESLAP	IBR	SEDGELAP	SURF
Bag	Det. Err.	Avg.	8.85	32.35	30.55	27.09	3.04	3.04	16.02	26.69	11.95
		Med.	8.45	4.76	4.76	4.79	3.09	3.09	5.72	4.80	20.69
	Aff. Err.	Avg.	0.16	0.19	0.27	0.27	0.25	0.20	0.25	0.25	0.09
		Med.	0.16	0.13	0.21	0.21	0.25	0.21	0.19	0.19	0.07
Books	Det. Err.	Avg.	17.96	111.14	104.08	108.20	46.24	46.39	13.61	70.27	142.19
		Med.	9.26	34.70	30.75	32.73	17.70	17.76	7.12	13.57	161.16
	Aff. Err.	Avg.	0.13	0.20	0.30	0.30	0.14	0.11	0.22	0.33	0.11
		Med.	0.11	0.12	0.19	0.19	0.10	0.10	0.15	0.18	0.10
Cube	Det. Err.	Avg.	9.08	30.73	29.89	30.02	28.65	28.65	8.22	42.09	25.47
		Med.	7.15	11.12	12.62	10.83	23.50	23.50	5.39	13.63	15.19
	Aff. Err.	Avg.	0.55	0.86	0.93	0.92	0.27	0.25	0.69	0.84	0.27
		Med.	0.46	0.48	0.52	0.52	0.31	0.30	0.59	0.43	0.36
PlushDog	Det. Err.	Avg.	11.38	26.60	23.57	28.62	10.50	11.18	15.58	24.09	16.19
		Med.	7.70	5.7	6.19	5.83	6.59	6.58	8.20	6.51	5.94
	Aff. Err.	Avg.	0.23	0.21	0.26	0.25	0.20	0.17	0.23	0.34	0.18
		Med.	0.25	0.15	0.20	0.20	0.15	0.13	0.19	0.21	0.10
Poster	Det. Err.	Avg.	11.60	51.33	41.78	45.71	18.60	19.74	11.86	46.27	10.92
		Med.	7.55	5.30	5.21	5.35	4.94	4.98	5.37	5.45	5.66
	Aff. Err.	Avg.	0.15	0.15	0.24	0.25	0.14	0.11	0.25	0.28	0.10
		Med.	0.10	0.09	0.17	0.17	0.09	0.09	0.16	0.16	0.09
Flacon	Det. Err.	Avg.	14.75	70.39	62.64	66.93	61.34	61.38	14.56	65.30	100.98
		Med.	7.48	15.59	14.04	14.07	22.00	21.67	7.4	10.44	86.44
	Aff. Err.	Avg.	0.10	0.17	0.27	0.27	0.16	0.09	0.21	0.32	0.07
		Med.	0.08	0.11	0.18	0.18	0.10	0.06	0.14	0.17	0.06
T-Rex	Det. Err.	Avg.	27.08	26.43	29.04	30.50	23.43	27.64	13.20	36.40	60.86
		Med.	39.71	6.75	6.46	6.68	7.20	7.50	8.05	6.33	27.81
	Aff. Err.	Avg.	0.12	0.21	0.28	0.29	0.17	0.13	0.27	0.34	0.11
		Med.	0.13	0.12	0.19	0.19	0.10	0.08	0.21	0.19	0.09

Table 3: The table contains the number of features on the test cases (rows) detected by a detectors (columns). The matching is done based on the GT. 'Det. Err.' is computed, where there is no feature in the 3 pixel radius of the GT, otherwise the affine error('Aff. Err.') can be computed.

		EBR	HARAFF	HARHES	HARLAP	HESAFF	HESLAP	IBR	SEDGELAP	SURF
Bag	# All	22.32	1149.26	1173.00	1167.58	25.63	26.16	129.21	1292.42	8.89
	# Cant Rec	4.79	315.05	328.63	326.90	11.79	11.89	39.16	547.79	1.84
	# Det. Err	15.79	119.05	122.90	122.21	6.68	6.68	44.95	123.84	2.16
	# Aff. Err	4.26	728.84	735.95	733.00	9.47	9.89	48.42	636.32	5.63
Books	# All	309.74	7839.11	8246.84	7992.05	466.79	468.47	730.84	8699.26	152.68
	# Cant Rec	191.32	4714.26	4958.89	4813.16	285.79	287.37	463.53	4819.00	87.11
	# Det. Err	78.26	270.58	310.05	284.95	49.95	49.95	106.42	718.37	7.63
	# Aff. Err	54.21	2953.00	3067.95	2994.32	135.47	135.47	172.47	3221.68	58.32
Cube	# All	89.42	804.90	830.32	811.42	50.16	50.16	32.21	1663.21	8.05
	# Cant Rec	106.74	466.21	475.63	470.47	20.89	20.89	13.05	856.16	4.89
	# Det. Err	7.37	69.37	80.05	69.89	23.37	23.37	10.16	309.63	1.95
	# Aff. Err	7.37	299.63	304.11	301.37	7.32	7.32	10.74	518.26	1.89
PlushDog	# All	34.42	1401.47	1604.84	1419.53	326.74	336.05	127.58	3612.26	39.26
	# Cant Rec	12.32	669.58	768.05	682.32	175.16	181.42	55.42	1908.95	19.53
	# Det. Err	21.53	191.16	240.00	192.58	79.00	79.37	50.84	504.16	9.00
	# Aff. Err	3.79	557.74	613.63	561.00	79.74	81.89	25.32	1228.42	11.89
Poster	# All	346.16	5945.89	6915.32	5998.00	1522.26	1544.47	591.47	13820.60	306.63
	# Cant Rec	186.00	2461.42	2902.79	2496.79	645.47	662.58	325.90	7316.05	157.74
	# Det. Err	110.05	157.16	218.90	159.16	68.37	69.74	96.79	803.84	18.42
	# Aff. Err	57.32	3337.47	3797.89	3351.37	812.95	816.90	176.21	5741.37	133.84
Flacon	# All	546.84	3062.00	3980.58	3119.42	1300.53	1355.63	286.68	7429.58	283.42
	# Cant Rec	245.16	1622.74	2122.11	1644.11	767.21	796.42	133.26	3543.16	164.79
	# Det. Err	170.21	152.58	194.74	154.47	75.21	76.32	77.42	813.16	22.68
	# Aff. Err	163.95	1370.47	1791.58	1408.26	533.84	561.32	87.26	3307.68	112.32
T-Rex	# All	9.58	958.79	1048.74	973.53	150.37	155.63	77.84	2377.21	33.53
	# Cant Rec	4.26	425.95	464.37	439.16	52.11	57.21	24.74	1593.21	11.95
	# Det. Err	6.89	189.42	208.53	189.90	48.00	48.00	35.42	289.26	10.32
	# Aff. Err	1.32	371.63	404.53	372.58	56.11	56.11	21.84	527.32	12.74

Table 4: The table shows the average (Avg.) and median (Med.) error of the detectors, when the feature matching is done based on the GT location of the features. Det. Err. and Aff. Err are given by the reprojection error and the Frobenius norm of difference matrix of retrieved and GT affine matrices, respectively.

			EBR	HARAFF	HARHES	HARLAP	HESAFF	HESLAP	IBR	SEDGELAP	SURF	
Bag	Det. Err.	Avg.	27.57	15.40	15.58	15.63	135.31	135.31	16.14	12.19	101.31	
		Med.	16.47	6.78	6.92	6.88	174.39	174.39	7.51	5.61	152.89	
	Aff. Err.	Avg.	0.20	0.44	0.51	0.51	0.46	0.35	0.40	0.52	0.14	
		Med.	0.20	0.29	0.34	0.34	0.49	0.32	0.27	0.33	0.09	
	Books	Det. Err.	Avg.	17.22	7.81	8.08	7.75	21.18	21.18	13.68	6.47	44.43
			Med.	8.50	5.18	5.20	5.12	10.39	10.39	7.75	4.61	21.66
Aff. Err.		Avg.	0.37	0.43	0.53	0.53	0.20	0.16	0.33	0.86	0.12	
		Med.	0.14	0.25	0.33	0.33	0.13	0.11	0.19	0.36	0.10	
Cube		Det. Err.	Avg.	70.92	13.48	14.64	13.74	28.57	28.57	46.87	8.22	165.84
			Med.	68.46	6.30	6.82	6.26	14.60	14.60	20.91	5.33	187.73
	Aff. Err.	Avg.	0.95	3.24	3.27	3.26	1.42	1.39	1.28	3.88	1.65	
		Med.	0.77	0.77	0.81	0.80	0.65	0.64	0.66	0.88	0.43	
PlushDog	Det. Err.	Avg.	25.32	12.26	11.28	12.40	11.11	11.08	16.80	6.98	22.03	
		Med.	14.64	5.97	5.99	5.99	6.30	6.32	9.18	4.82	14.17	
	Aff. Err.	Avg.	0.42	0.65	0.67	0.68	0.32	0.29	0.39	1.55	0.26	
		Med.	0.44	0.29	0.33	0.33	0.23	0.18	0.27	0.50	0.12	
Poster	Det. Err.	Avg.	15.29	6.55	6.29	6.54	9.21	9.09	11.67	4.88	10.87	
		Med.	10.17	4.97	4.94	4.96	5.42	5.32	6.49	4.04	7.11	
	Aff. Err.	Avg.	0.34	0.34	0.43	0.43	0.23	0.20	0.35	0.85	0.11	
		Med.	0.12	0.21	0.28	0.28	0.13	0.10	0.22	0.33	0.09	
Flacon	Det. Err.	Avg.	10.57	11.35	9.90	11.28	12.82	12.69	13.24	6.06	19.22	
		Med.	6.80	6.03	5.65	6.00	8.28	8.23	8.25	4.46	13.51	
	Aff. Err.	Avg.	0.40	0.40	0.54	0.51	0.30	0.21	0.33	1.14	0.09	
		Med.	0.12	0.21	0.30	0.30	0.17	0.09	0.20	0.39	0.06	
T-Rex	Det. Err.	Avg.	584.59	13.79	13.43	13.78	26.65	26.65	12.76	9.99	23.99	
		Med.	577.87	8.67	8.25	8.68	21.22	21.22	7.80	5.69	12.28	
	Aff. Err.	Avg.	0.23	0.70	0.74	0.74	0.37	0.29	0.50	1.11	0.13	
		Med.	0.29	0.35	0.40	0.40	0.23	0.15	0.34	0.46	0.09	