

IGFTT: towards an efficient alternative to SIFT and SURF

Ícaro Oliveira de Oliveira
Federal Technological
University of Paraná
Brazil
icaroo@utfpr.edu.br

Keiko Veronica Ono
Fonseca
Federal Technological
University of Paraná
Brazil
keiko@utfpr.edu.br

Eduardo Todt
VRI Research Group,
Federal University of
Paraná
Brazil
todt@inf.ufpr.br

ABSTRACT

The invariant feature detectors are essential components in many computer vision applications, such as tracking, simultaneous localization and mapping (SLAM), image search, machine vision, object recognition, 3D reconstruction from multiple images, augmented reality, stereo vision, and others. However, it is very challenging to detect high quality features while maintaining a low computational cost. Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) algorithms exhibit great performance under a variety of image transformations, however these methods rely on costly keypoint's detection. Recently, fast and efficient variants such as Binary Robust Invariant Scalable Keypoints (BRISK) and Oriented Fast and Rotated BRIEF (ORB) were developed to offset the computational burden of these traditional detectors.

In this paper, we propose to improve the Good Features to Track (GFTT) detector, coined IGFTT. It approximates or even outperforms the state-of-art detectors with respect to repeatability, distinctiveness, and robustness, yet can be computed much faster than Maximally Stable Extremal Regions (MSER), SIFT, BRISK, KAZE, Accelerated KAZE (AKAZE) and SURF. This is achieved by using the search of maximal-minimum eigenvalue in the image on scale-space and a new orientation extraction method based on eigenvectors.

A comprehensive evaluation on standard datasets shows that IGFTT achieves quite a high performance with a computation time comparable to state-of-the-art real-time features. The proposed method shows exceptionally good performance compared to SURF, ORB, GFTT, MSER, Star, SIFT, KAZE, AKAZE and BRISK.

Keywords

IGFTT, feature detectors, keypoint, computer vision, repeatability

1 INTRODUCTION

Feature detectors have become an essential component in contemporary computer vision research. The main goal is to find salient image keypoints that can be repeatedly detected under various image transformations and then construct distinctive and robust representations for them.

This paper aims to tackle this problem by developing an improvement on the well-known GFTT keypoint detector [21]. This technique is attractive because of its good performance. For feature detection, scale invariant and stable keypoints are selected in the scale space according to maximal-minimum eigenvalues responses. Furthermore, it was built a fast accurate orientation estimation by the eigenvector's orientation.

We have validated the IGFTT detector whose precision, recall and the execution time figures. The experiments show that IGFTT achieves quite a high performance with a computation time comparable to state of the art, e.g. ORB and it is more efficient in terms of repeatability than KAZE, BRISK, SURF and SIFT. Also, the orientation extraction method developed in this paper presented less errors than other methods. The rest of this paper is organized as follows: First, an overview of related work is given in Section 2. Section 3 describes a review about GFTT detector. Section 4 describes the improvement to GFTT developed here. The experimental evaluation is carried out in Section 5, and finally Section 6 presents the discussion and conclusions.

2 BACKGROUND

2.1 Feature Detection

Corners are points in an image where two lines meet perpendicularly [10]. They can be any points between two lines with different directions or between two points with strong image gradients. Corners are particularly important since they can be used to locate

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

and make the registration of objects and to provide measures of their dimensions, for example, knowledge about orientation will be vital for a robot finds the best way of picking up an object, inspection and other applications [8].

Thereby, corners are interesting for identifying parts of an image, efficient to match image to image and have great relevance to the accuracy and efficiency of machine vision. The corners with local maximum or minimum intensity are called interest points or keypoints.

SIFT [12] is a pioneering method which produces high quality features based in gradients and requires high computational effort. SURF [5] detects keypoints faster than the original SIFT, without loss in performance and, recently, other methods were proposed improving either processing time or repeatability, e.g. ORB [20], KAZE [4] and BRISK [23].

SIFT [12], SURF [5], BRISK [23] and ORB [20] have the same structure: they detect best keypoints in a scale pyramid type like shown in Figure 1 and extract orientation using the directed gradients or moments.

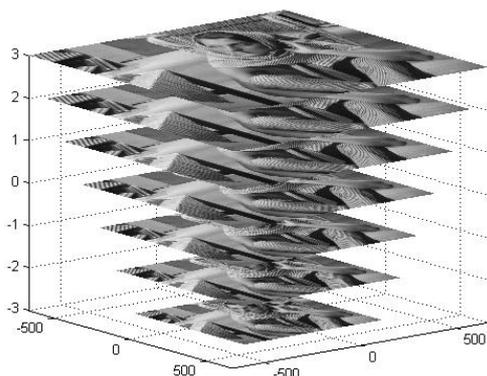


Figure 1: Image Pyramid [9]

Each detector applies an algorithm to detect stable keypoints in each scale level. For example, SIFT [12] detects in a Difference of Gaussians applied in each level, SURF [5] detects in wavelet and integral images, BRISK [23] detects in a scale space with interpolation and ORB [20] detects in a simple pyramid scales. Other detectors like AGAST [13], FAST [18] and SUSAN [22] detect stable keypoints in an unique image and don't use scale pyramids or extract orientation.

SIFT [12] extracts scale-invariant features through detecting local extremas of the Difference-of-Gaussian (DoG) over scale space applied in the image. The DoG is a faster approximation of Laplacian of Gaussian (LoG). SURF [5] proposes the Fast-Hessian detector. This method approximates the second order Gaussian derivatives of Hessian matrix from rectangle filters. Furthermore, the Fast-Hessian detector works with

integral images to compute 3 times faster than DoG. Recently, Features from Accelerated Segment Test (FAST) [18] detector is based on the SUSAN corner detector [22]. The circular area center is used to determine brighter and darker neighboring pixels on the circle describing the segment. FAST uses a Bresenham's circle [18] of diameter 3.4 pixels as test mask. Thus, for a full accelerated segment test, 16 pixels have to be compared to obtain the value of the nucleus. In the sequence, FAST selects the 9 top segments (FAST-9) that get high repeatability between 16 pixels. For each keypoint, the FAST calculates a sum of the differences between each 9 circular pattern's pixels and the nucleus' pixel and selects the keypoint with a sum lower than a threshold. This parameter controls the sensitivity of the corner response. A large t-value results in few but therefore only strong corners, while a small t-value yields also corners with smoother gradients. The alternative FAST-ER [19] generalizes FAST by allowing it to learn the 9 top segments with the best repeatability and small loss of efficiency for any scene.

AGAST [13] improves FAST performance by combining specialized decision trees. Since these FAST-based detectors do not deal with scale change, BRISK [23] takes AGAST to detect feature candidates and searches for the maximum FAST score over scale space to achieve scale invariance. KAZE [4] proposes an automatic feature detection in nonlinear scale spaces using efficient AOS techniques and variable conductance diffusion. Next, AKAZE [17] proposes to use recent numerical schemes called Fast Explicit Diffusion (FED) embedded in a pyramidal framework to dramatically speed-up feature detection in nonlinear scale spaces. The STAR [2] keypoint detector uses an approximation that allows to preserve rotational invariance applied in scale space formed by a bi-level approximation of the Laplacian of Gaussians (LoG) filter. ORB [20] detects FAST features filtered by Harris at each level in the scale pyramid of the image and calculates a new fast and accurate orientation component to FAST. A Maximally Stable Extremal Region (MSER) [14] is a connected component of an appropriately thresholded image. The word 'extremal' refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary. The 'maximally stable' in MSER describes the property optimized in the threshold selection process.

2.2 Feature Description

SIFT [12] creates a histogram of local gradient orientations and locations, where the gradient orientations are quantized into 8 orientation bins and the space locations are quantized into a 4x4 grid.

SURF [5] works with the Haar wavelet responses as the local features. Since the above gradient-based descriptors can only deal with linear illumination changes, some other methods have been proposed to tackle more general illumination changes by using relative intensity orders of pixels rather than the original intensities. BRIEF [7] uses a relatively small number of binary tests between pixels to represent the local patch as a binary string. ORB [20] develops a rotation invariant by the intensity centroid that extends the descriptor BRIEF, and it is more discriminative by learning a good subset of binary tests than BRIEF original. BRISK [23] introduces a Gaussian weighted pattern for sampling the neighborhood of keypoints. The long-distance pairs are used to estimate the local dominant orientation and the short-distance pairs are used to build a binary descriptor. FREAK [1] proposes a retinal sampling pattern based on the human visual system, and computes the binary descriptor by comparing image intensities over the retinal patterns. KAZE [4] uses the M-SURF descriptor adapted to a nonlinear scale space framework. AKAZE [17] introduces a Modified-Local Difference Binary (M-LDB) descriptor that is highly efficient, exploits gradient information from the nonlinear scale space, is scale and rotation invariant and has low storage requirements.

In practice, it is very challenging to obtain a high quality feature whilst maintaining a low computational cost on several transformations. Therefore, this work aims to improve the GFTT detector developing a novel orientation estimation and creating by simple scale pyramid to GFTT detector.

3 GOOD FEATURES TO TRACK: REVIEW

In this section the Good Features to Track [21] will be briefly described. Consider an image sequence $I(x,t)$, with $x = [u,v]^T$ where u and v are the coordinates of an image point. If a point of time sampling t is substantially high, then the points of the image I are displaced, however their intensities remain unchanged:

$$I(x,t) = I(\delta(x),t + \tau) \quad (1)$$

where $\delta(\cdot)$ is the motion field that specifies the transformation applied to image points. The authors approximate the transformation to a translation through the fast-sampling hypothesis, in other words, $\delta(x) = x + d$, where d is a displacement vector. Variable d is used to search keypoints in the frame's sequences. The image motion model can keep some noises becoming not perfect, so the problem is to find d which minimizes the Sum of Squared Differences (SSD) to find the displacement d residuals, from this the equation below is computed:

$$\min_d \left(\sum_W [I(x+d,t+\tau) - I(x,t)]^2 \right) \quad (2)$$

where W is an image window around the keypoint and t is the frame in t time. If we apply first-order Taylor expansion of $I(x+d,t+\tau)$ into (2) we can obtain a simple linear system formed by

$$Gd = e \quad (3)$$

where

$$G = \sum_w \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix}, \quad e = -\tau \sum_w I_t [I_u \ I_v]^T$$

with $[I_u \ I_v] = \nabla I = [\frac{\partial I}{\partial u} \ \frac{\partial I}{\partial v}]$ and $I_t = \frac{\partial I}{\partial t}$. From Eq. (3): d is the solution of (3), that is, $d = G^{-1}e$, and is used to predict a new (registered) frame. This method is iterated through of the Newton-Raphson scheme to converge the displacement d . So, assuming that λ_1 and λ_2 are the eigenvalues of G , the feature is detected if $\min(\lambda_1, \lambda_2) > \lambda$; where λ is an user-defined threshold.

Thereafter, it is assuming the image I and next image J . If I and J are dissimilar images then the feature is dropped. The dissimilarity images I and J is measured by the equation

$$\text{sum}(I - J) > \text{threshold} \quad (4)$$

Between consecutive frames a translation model is sufficient for tracking, however an affine model is necessary when frames are far.

It is more expensive to calculate an affine model for each frame than use a simple model, for example, a scale model. In this case, the IGFTT used a simple scale model and obtains better results related to precision than SIFT, SURF, AKAZE, KAZE, FAST, ORB, MSER, STAR and GFTT.

4 IMPROVE GOOD FEATURES TO TRACK (IGFTT)

Focusing on computation efficiency, our detection methodology is inspired by the work of Shi Tomasi et al. [21] for detecting keypoints in the image. Aiming the achievement of invariance to scale which is crucial for high-quality keypoints, we go a step further by searching for keypoint not only in the image plane, but also in scale-space.

The IGFTT has the same GFTT's parameters like *minDistance*, *qualityLevel*, *blockSize* and N . *minDistance* filters the keypoints with maxima-minimal eigenvalues lower than the *minDistance*.

Variable *qualityLevel* is used to threshold the minimum eigenvalues. In other words, the maximum global eigenvalue is multiplied by the variable *qualityLevel*. We select the eigenvalues smaller than *qualityLevel*. Variable *blockSize* is the block's length used to calculate the minimum eigenvalues in the image. Variable *N* is maxima amount of detected keypoints. Further, we add the variables *scaleLevels* and *scaleFactor* to generate the scale space. Variable *scaleLevels* is the amount of levels in the scale space and variable *scaleFactor* is the applied factor in the image at each level of the scale space.

In the literature, there are three scale-space types: wavelet, gaussian pyramid and simple pyramid. The wavelet is linear scale space representation that analyzes the signal from scale and resolution. A Gaussian pyramid is a scale-space that creates a set of images scaled down from a image. Each image is also weighted by a Gaussian blur. This pyramid is used to create a scale space with blur invariant, however the eigenvalues already are blur invariants. The simple pyramid is a scale-space that creates a set of scaled down images from an original image. In this case, we used the simple pyramid to make easier implementation.

Data: the pyramid scale space from images with various scales

Result: KeyPoints

initialization;

```

foreach image in pyramid do
    calculate gradient matrix;
    calculate the covariance in the gradient matrix;
    extract eigenvectors and eigenvalues in the
    directions x and y;
    select the minimal eigenvalues between directions
    x and y and the eigenvector for each pixel;
    calculate the maximum eigenvalue local;
    select the keypoints with eigenvalues bigger than
    maximum value local * qualityLevel;
    sort the selected keypoints by the eigenvalue;
    foreach point in keypoints do
        if total_KeyPoints == N then
            | break;
        end
        if distance between point and all keypoints <
        minDistance then
            | KeyPoints.add(point);
            | total_KeyPoints = total_KeyPoints + 1;
        end
    end
end
    
```

Algorithm 1: IGFTT

4.1 Fast Orientation by Eigenvector

Our approach uses a simple but effective measure of corner orientation, based on the eigenvector, because the eigenvector represents the direction preserved by a linear transformation applied in the gradient matrix. From $\min(\lambda_1, \lambda_2)$ we selected the eigenvector of G with the minimal eigenvalue. The coordinates (x, y) of the eigenvector are used to extract the interest point orientation.

$$\text{angle} = \text{atan2}(y, x); \quad (5)$$

where atan2 is the quadrant-aware version of arctan.

4.2 Descriptor

We use the FREAK descriptor applied in each scale space. For a detected keypoint k at scale σ_i , the FREAK descriptor extracts the pattern formed by Difference-of-Gaussians inspired in the retinal pattern of the eye. Furthermore, FREAK extracts this pattern in various scales and orientations. the FREAK has integral images to accelerate the description process. Finally, the descriptor is interpolated and normalized.

5 EXPERIMENTS

We have evaluated our method on the standard Oxford dataset [15] and on two new datasets [24, 11] with geometric and photometric transformations like rotation, scale, viewpoint, image blur, JPEG compression and illumination. The implementations for all detectors came from OpenCV 3.0 beta [6].

5.1 Detector evaluation

We compare IGFTT detector with SURF, ORB, GFTT, MSER, STAR, SIFT, KAZE, AKAZE and BRISK detectors. We test the precision and recall scores of different detectors for gradually increasing transformation. In these experiments we set $\text{minDistance} = 1$. Through repeatability criterion introduced in [16], we analyze the overlap error of two correspond regions in various transformations. This error is defined as

$$1 - \frac{R_{\mu_a} \cap R_{H^T \mu_b H}}{R_{\mu_a} \cup R_{H^T \mu_b H}} \quad (6)$$

where R_{μ} represents the elliptic region defined by $x^T \mu x = 1$. H is the homography between the two images. The intersection of the regions is $R_{\mu_a} \cap R_{H^T \mu_b H}$ and $R_{\mu_a} \cup R_{H^T \mu_b H}$ is their union. The intersection of the regions must be greater than 0. The area of these regions are computed numerically. The repeatability score for a given pair of images is defined as

$$\frac{\text{true keypoints}}{\text{detected keypoints}} \quad (7)$$

where the amount of keypoints with *overlap error* $< \varepsilon_t$ is represented by *true keypoints*. *Detected keypoints* is the amount of keypoints detected and the overlap error threshold represented by ε_t is 40%. We take into account only the regions located in scene's parts present in both images. In this case, the repeatability represents the recall score. The precision score is

$$\frac{\text{true keypoints}}{\text{matched keypoints}} \quad (8)$$

where matched keypoints are the amount of the keypoints with intersection of the regions greater than 0. The true keypoints are based in putative matches, in other words, a putative match is formed by a single pairing of keypoints, where a keypoint cannot be matched to more than one other. IGFTT detector consistently outperforms SURF, ORB, GFTT, MSER, STAR, SIFT, KAZE, and BRISK in most cases, which can be attributed for the minimal eigenvalues features used by the detector like shown in Figure 2 and Figure 3.

In this section, we compare the computation times of IGFTT to SURF, ORB, GFTT, HARRIS, MSER, Star, SIFT, KAZE, and BRISK. The experiments are carried out on desktop Intel Ivy Bridge Core i5-3450 3.10GHz CPU and 16GB DRAM, using the first image of the each dataset's sequence compared with other images. The results are averaged on 1000 experiments runs.

In general presented the IGFTT better averages than other detectors using minimal eigenvalues to select keypoints in each level of the scale-space. Furthermore, this feature, applied in each scale space level, is faster than the corresponding algorithms applied by the SURF, MSER, SIFT, AKAZE or KAZE. We used the average of the precision, recall and execution time compared to those obtained from the reference detectors [17, 4, 12, 5, 14, 3, 20, 23, 21].

The SIFT exhibits lower averages on precision (67.69% vs 93.21%) and recall (55.00% vs 81.78%), but more execution time (209.794ms vs 57.52ms) than IGFTT. The IGFTT eigenvalues and keypoints selection methods are faster and more repeatedly than Difference-of-Gaussian used by SIFT. The Difference-of-Gaussian used by SIFT did not detect the same keypoints of the first image in the other images from the various transformations in all datasets.

The SURF exhibits lower average on precision (85.13% vs 93.21%) and on recall (71.74% vs 81.78%) than IGFTT. The Hessian Matrix used by SURE, it did not detect the same keypoints of the first image in the other images from various transformations. In some cases,

the SURF shows better invariance than IGFTT in the datasets trees (blur) and venice (zoom), because of the Frobenius score and the box filter used by SURF. The Hessian Matrix applied in the integral images and Wavelet show more time values (83.61ms vs 57.52ms) than IGFTT, due to the complexity of both algorithms.

The KAZE shows lower average on precision (78.34% vs 93.21%) and on recall (69.57% vs 81.78%) than IGFTT. The nonlinear diffusion filtering did not detect the same keypoints of the first image in the other images with various transformations. Furthermore, KAZE is more complex than IGFTT, the KAZE showing a more average time (409.64ms vs 57.52ms) than IGFTT.

The AKAZE shows lower average on precision lower (78.52% vs 93.21%) and on recall (65.16% vs 81.78%) than IGFTT. The Fast Explicit Diffusion filtering did not detect the same keypoints of the first image in the other images from various transformations. Furthermore, the AKAZE is more complex than IGFTT, showing more average time (119.08ms vs 57.52ms) than IGFTT.

The BRISK shows lower average on precision (78.95% vs 93.21%) and on recall (58.21% vs 81.78%) than IGFTT. The pyramid with octave levels and octave in-between levels did not detect the same keypoints of the first image in the other images from various transformations. But, BRISK is faster (23.15ms vs 57.52ms) than IGFTT in average of the execution time, because it has AGAST detector that is more efficient than GFTT.

The ORB presents lower average on precision (88.30% vs 93.21%) and on recall (78.46% vs 81.78%) than IGFTT. The Harris score used by ORB did not detect the same keypoints of the first image in the other images from various transformations. In some cases ORB shows better invariance than IGFTT in the Reichstag (various transformations) and ubc (jpeg compression) datasets, due to the Harris score. But, the ORB is faster (16.58ms vs 57.52ms) than IGFTT in average of the time, because the FAST detector is more efficient than GFTT detector.

The GFTT shows lower average on precision (74.71% vs 93.21%) and on recall (55.81% vs 81.78%) than IGFTT. The GFTT detector is not invariant scale, viewpoint, rotation and zoom. In other words, the GFTT detector did not detect the same keypoints of the first image in the other images with various transformations. But, the GFTT is faster (19.38ms vs 57.52ms) than IGFTT in average time.

The STAR shows lower average on precision (71.90% vs 93.21%) and on recall (53.80% vs 81.78%) than IGFTT. The Star detector is not fully invariant to scale, viewpoint, rotation and zoom. In other words, the STAR detector did not detect the same keypoints of the first image in the other images with various

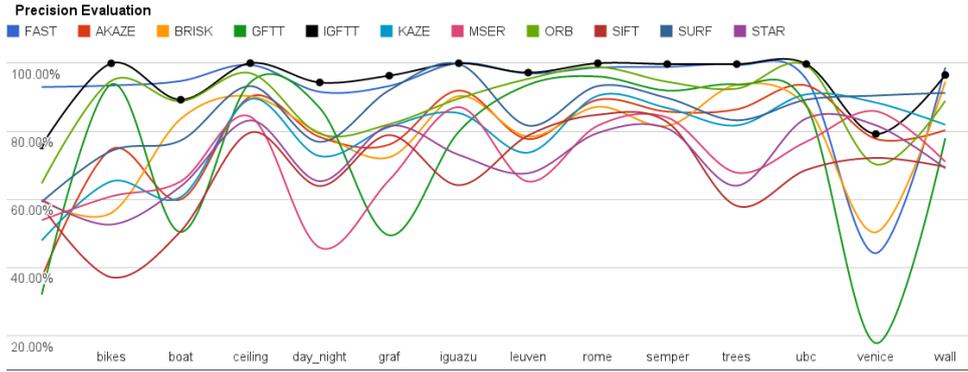


Figure 2: The precision of the detectors on various datasets with affine transformations.

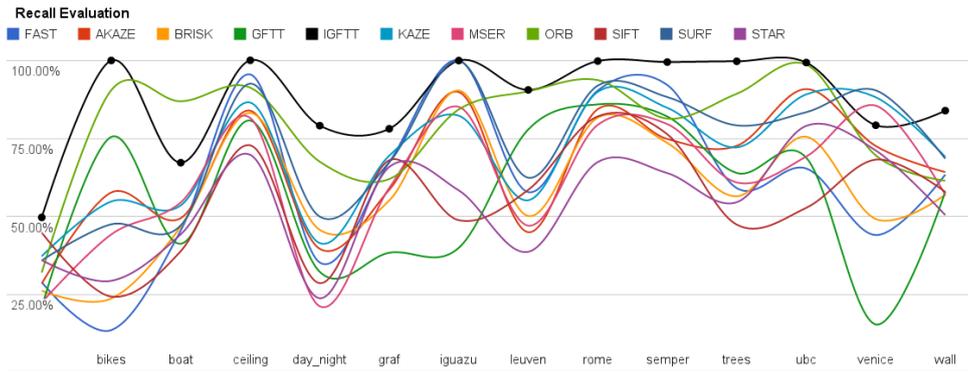


Figure 3: The recall of the detectors on various datasets with affine transformations.

transformations. But, the STAR is faster (18.52ms vs 57.52ms) than IGFTT in average time.

The MSER shows lower average on precision (71.14% vs 93.21%) and on recall (60.51% vs 81.78%) than IGFTT. The MSER methods to detect regions are based on extremal regions did not detect the same keypoints of the first image in the other images with various transformations, although the detector been fully invariant to these transformations. Furthermore, the MSER was more complex than IGFTT, the MSER was more average time (89.88ms vs 57.52ms) than IGFTT.

The FAST shows lower average on precision (92.71% vs 93.21%) and on recall (61.34% vs 81.78%) than IGFTT. The FAST detector is not invariant to scale, viewpoint, rotation and zoom. In other words, the FAST detector did not detect the same keypoints of the first image in the other images with various transformations. But, the FAST is faster (4.73ms vs 57.52ms) than IGFTT in average time.

5.2 Descriptor evaluation

We compare IGFTT detector combined with SURF, ORB, SIFT, FREAK, BRIEF and BRISK descriptors with SURF, ORB, SIFT, KAZE and BRISK. Furthermore, we used the brute-force matching developed in OPENCV 3.0 to make the matching between the images. We used the precision and recall

scores. The overlap error threshold represented by ϵ_t is 50%. We take into account only the regions located in the part of the scene present in both images. This way, the recall score for a given pair of images is defined as

$$\frac{\text{true keypoints}}{\text{visible keypoints}} \tag{9}$$

where the amount of keypoints with *overlap error* $< \epsilon_t$ is represented by *true keypoints*, *visible keypoints* is the amount of keypoints in the part of the scene present in both images and the overlap error threshold represented by ϵ_t is 50%. In this case, the repeatability represents the recall score. The precision score is 8 where matched keypoints are the amount of the keypoints matched between the images pair. The true keypoints are based in putative matches, in other words, a putative match is formed by a single pairing of keypoints, where a keypoint cannot be matched to more than one other.

In this section, we compare the computation times between IGFTT combined with various descriptors SURF, ORB, SIFT, KAZE, and BRISK. The experiments are performed on desktop Intel Ivy Bridge Core i5-3450 3.10GHz CPU and 16GB DRAM, using the first image of the each dataset’s sequence compared with other images. The results are averaged on 1000 experiments runs. We used the average of the precision, recall and execution time to compare the all detectors and descriptors used here.

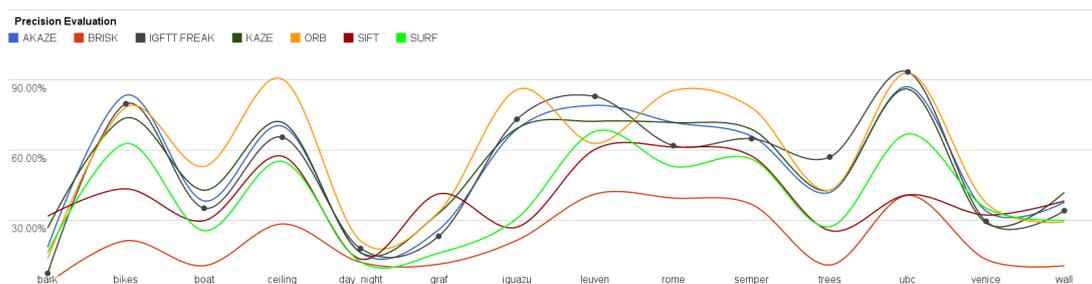


Figure 4: The precision of the detectors combined with descriptors on various datasets with affine transformations.

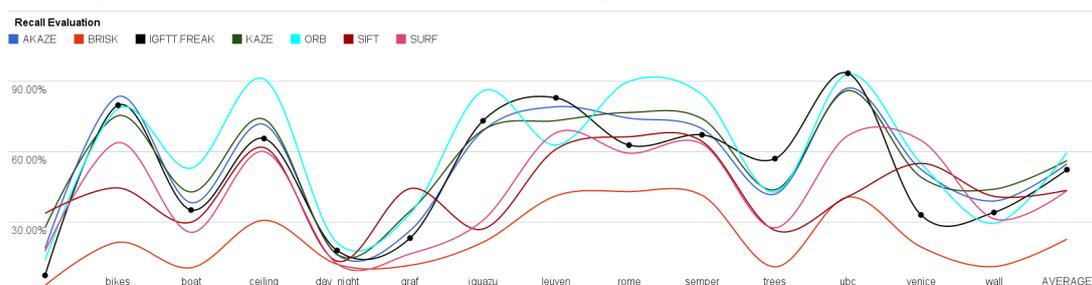


Figure 5: The recall of the detectors combined with descriptors on various datasets with affine transformations.

In some cases the IGFTT with FREAK descriptor (called IGFTT.FREAK) has better averages in precision and recall than SIFT, SURF and BRISK. Furthermore, IGFTT.FREAK is faster than SIFT, SURF, KAZE, AKAZE and BRISK.

The SIFT exhibits lower average on precision (40.02% vs 51.85%) and on recall (43.57% vs 52.34%), furthermore it is greater time average (957.85ms vs 64.77ms) than IGFTT.FREAK. The FREAK descriptor using retinal binary pattern describes very well the regions around the keypoints detected by the IGFTT and this combination show better results than the Difference-of-Gaussian and the descriptor based in gradients' orientations. Furthermore, in [1] the descriptor FREAK generate a binary string formed by a sequence of one-bit Difference-of-Gaussians. In other words, this descriptor is more discriminant than SIFT descriptor.

The SURF exhibits lower average on precision (39.66% vs 51.85%) and on recall (43.44% vs 52.34%) than IGFTT. The FREAK descriptor is faster and robust than SURF descriptor through cascade of binary strings. The Hessian Matrix applied in the integral images, Wavelet and shows more time average (657.60ms vs 64.77ms) than IGFTT.

The KAZE shows greater average on precision (53.31% vs 51.85%) and on recall (56.21% vs 52.34%) than IGFTT. The nonlinear diffusion filtering extracts some features more discriminant than the FREAK descriptor like shown in bark, boat, ceiling, graf, rome, semper and wall datasets. However, the KAZE is more complex than IGFTT, the KAZE shows more time average (629.55ms vs 64.77ms) than IGFTT.

The AKAZE shows lower average on precision (52.81% vs 51.85%) and on recall (54.82% vs 52.34%) than IGFTT. The Fast Explicit Diffusion filtering extracts some features more discriminant than the FREAK descriptor like shown in bark, bikes, boat, ceiling, graf, rome, semper, venice and wall datasets. However, the AKAZE is more complex than IGFTT, the AKAZE shows more time average (331.06ms vs 64.77ms) than IGFTT.

The BRISK shows lower average on precision (21.47% vs 51.85%) and on recall (22.75% vs 52.34%) than IGFTT. The descriptor BRISK based in a bit-string descriptor from intensity comparisons is lower discriminant than the retinal binary pattern used in the FREAK descriptor. Furthermore, BRISK was more complex than IGFTT, the BRISK shows more time average (74.33ms vs 64.77ms) than IGFTT.

The ORB results in greater average on precision (57.51% vs 51.85%) and on recall (59.59% vs 52.34%) than IGFTT. The Oriented BRIEF descriptor and the ORB detector are more discriminant in some cases than IGFTT with FREAK descriptor. The ORB shows a better invariance than IGFTT in the bark, boat, ceiling, day_night, graf, iguazu, rome, semper and venice datasets. Furthermore, the ORB is faster (24.05ms vs 64.77ms) than IGFTT in time average, the FAST detector used in the ORB is faster than IGFTT detector.

6 CONCLUSIONS

We have presented a novel method named IGFTT, which tackles the classic Computer Vision problem of detecting image keypoints for cases without sufficient a priori knowledge on the scene, camera poses

and transformation. In contrast to well-established algorithms with proven high performance, such as SIFT and SURF, the method at hand offers a dramatically faster alternative at comparable precision performance - a statement which we base on an extensive evaluation using an established framework.

IGFTT relies on an easily configurable, the unique properties of IGFTT can be useful for a wide spectrum of applications, in particular for tasks with hard real-time constraints or limited computation power: IGFTT finally offers the quality of high-end keypoints in such time-demanding applications.

Amongst avenues for further research into IGFTT, we aim to explore alternatives to the GFTT search to yield higher repeatability whilst maintaining or improve the speed. Furthermore, we aim at analyzing both theoretically and experimentally other efficient detectors applied in the scale space or other filters like ORB applied.

7 REFERENCES

- [1] R. O. A. Alahi and P. Vanderghenst. Freak: Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2012.
- [2] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. *European Conference on Computer Vision 2008*, 2008.
- [3] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In D. A. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008.
- [4] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI, ECCV'12*, pages 214–227, 2012.
- [5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision—ECCV 2006*, pages 404–417. Springer, 2006.
- [6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [7] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.
- [8] E. R. Davies. *Computer and machine vision theory, algorithms, practicalities*. Elsevier, Waltham, Mass., 2012.
- [9] C. R. del Blanco Adán. Computer vision, 2014. [Online; accessed em 01-April-2014].
- [10] K. Demaagd, A. Oliver, N. Oostendorp, and K. Scott. *Practical Computer Vision with SimpleCV - Making Computers See in Python*. O'Reilly Media, 2012.
- [11] E. D. J. Heiny and J. M. Frahm. Comparative evaluation of binary features. *European Conference on Computer Vision*, 2012.
- [12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.
- [13] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*, September 2010.
- [14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Proc. of British Machine Vision Conference*, pages 384–396, 2002.
- [15] K. Mikolajczyk. Affine covariant features, 2014. [Online; accessed em 01-Abril-2014].
- [16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, Nov. 2005.
- [17] J. N. Pablo F. Alcantarilla and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *British Machine Vision Conference (BMVC)*, 2013.
- [18] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.
- [19] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence*, pages 105–119, 2010.
- [20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. *Proceedings of the 2011 International Conference on Computer Vision*, 2011.
- [21] J. Shi and C. Tomasi. Good features to track. *Computer Vision and Pattern Recognition*, 1994.
- [22] S. M. Smith and J. Brady. Susan—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
- [23] M. C. Stefan Leutenegger and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [24] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. *Computer Vision and Pattern Recognition*, 2008.