

# Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces

Pablo F. Alcantarilla  
pablo.alcantarilla@gatech.edu

Jesus Nuevo  
jesus.nuevochiquero@gmail.com

Adrien Bartoli  
adrien.bartoli@gmail.com

School of Interactive Computing  
Georgia Institute of Technology  
Atlanta, GA, USA

TrueVision Solutions  
POBox 191,  
Toowong, QLD, Australia

ISIT-UMR 6284 CNRS  
Université d'Auvergne  
Clermont-Ferrand, France

---

## Abstract

We propose a novel and fast multiscale feature detection and description approach that exploits the benefits of nonlinear scale spaces. Previous attempts to detect and describe features in nonlinear scale spaces are highly time consuming due to the computational burden of creating the nonlinear scale space. In this paper we propose 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. In addition, we introduce 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. We present an extensive evaluation that shows the excellent compromise between speed and performance of our approach compared to state-of-the-art methods such as BRISK, ORB, SURF, SIFT and KAZE.

## 1 Introduction

The best known multiscale feature detection and description approaches are SIFT [17] and SURF [4], SURF being less computationally demanding than SIFT. Both approaches make use of the Gaussian scale space, either by constructing the Gaussian scale space in a pyramidal framework such as in SIFT, or by approximating Gaussian derivatives through box filters as in SURF. This has important drawbacks since Gaussian blurring does not preserve object boundaries and smoothes to the same extent details and noise at all scales, spoling localization accuracy and distinctiveness.

To overcome this problem, some methods have been presented recently [2, 19] that aim to detect and describe features in nonlinear scale spaces. This makes blurring locally adaptive to the image data, blurring small details but preserving object boundaries. KAZE features were introduced in [7]. This method increases repeatability and distinctiveness with respect to SIFT and SURF thanks to the use of nonlinear diffusion filtering. The main drawback of KAZE is that it is computationally intense. Since there are no analytical solutions

to solve the nonlinear diffusion equation, one needs to use numerical methods to approximate the solution. In KAZE, this is done by means of Additive Operator Splitting (AOS) schemes [21]. While AOS schemes are absolutely stable for any time step and easily parallelizable, they require solving a large system of linear equations to obtain a solution. BFSIFT features [19] are detected by replacing the standard Gaussian scale space performed by SIFT by an anisotropic scale space using bilateral filtering. BFSIFT improves over SIFT in Synthetic Aperture Radar (SAR) image matching problems where images are corrupted by speckle noise. However, the price to pay is the higher computational complexity of the bilateral filter for building the nonlinear scale space, making this approach prohibitive for real-time image matching applications.

With the proliferation of camera-enabled mobile devices that have limited computational resources, new features have appeared that aim to reduce computational complexity while keeping up to the performance of methods such as SIFT and SURF. ORB [16] and BRISK [11] speed-up feature detection and description by combining modifications of the FAST corner detector [15] and binary descriptors based on BRIEF [5] with scale and rotation invariance. ORB and BRISK features are much faster to compute than SIFT and SURF, while showing comparable performance mainly for small image transformations.

In this paper we aim to obtain low-computationally demanding features taking advantage of the benefits of nonlinear diffusion filtering. For this purpose, we introduce a recent mathematical framework called Fast Explicit Diffusion (FED) [7, 8] to feature detection and description problems. By means of FED schemes, a nonlinear scale space can be built much faster than with any other kind of discretization scheme. Furthermore, FED schemes are extremely easy to implement and are more accurate than AOS schemes. We propose to embed FED schemes in a pyramidal framework with a fine to coarse strategy to speed-up dramatically feature detection in nonlinear scale spaces. We call these features *Accelerated-KAZE* (A-KAZE) due to the dramatic speed-up introduced by FED schemes embedded in a pyramidal framework.

To preserve low computational demand and storage requirement, we also introduce a highly efficient Modified-Local Difference Binary (M-LDB) descriptor. While the original LDB descriptor presented in [22] is neither rotation nor scale invariant as BRIEF is, we build a robust binary descriptor that is rotation and scale invariant and exploits gradient information from the nonlinear scale space, increasing distinctiveness. Our novel A-KAZE features are faster to compute than SURF, SIFT and KAZE and also exhibit much better performance in detection and description than previous methods, including ORB and BRISK.

## 2 Nonlinear Diffusion Filtering

### 2.1 Background

Nonlinear diffusion filtering describes the evolution of the luminance of an image through increasing scale levels as the divergence of a certain *flow* function that controls the diffusion process. These approaches are described by nonlinear partial differential equations (PDEs), due to the nonlinear nature of the involved differential equations that diffuse the luminance of the image through the nonlinear scale space. The classic nonlinear diffusion equation is

$$\frac{\partial L}{\partial t} = \text{div}(c(x,y,t) \cdot \nabla L) , \quad (1)$$

where  $\text{div}$  and  $\nabla$  are respectively the divergence and gradient operators, and  $L$  is the image luminance. Thanks to the introduction of a *conductivity* function ( $c$ ) in the diffusion equation, it is possible to make the diffusion adaptive to the local image structure. The function  $c$  depends on the local image differential structure, and this function can be either a scalar or a tensor. Time  $t$  is the scale parameter, and larger values lead to simpler image representations. In anisotropic diffusion the image gradient magnitude controls the diffusion at each scale level. This way, the conductivity function  $c$  is defined as:

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|), \quad (2)$$

where the function  $\nabla L_\sigma$  is the gradient of a Gaussian smoothed version of the original image  $L$ . We consider one of the two conductivity functions introduced in the seminal work of Perona and Malik [14], although other conductivity functions are also possible. The conductivity function  $g_2$  promotes wide regions over smaller ones:

$$g_2 = \frac{1}{1 + \frac{|\nabla L_\sigma|^2}{\lambda^2}} \quad (3)$$

The parameter  $\lambda$  is the contrast factor that controls the level of diffusion. It determines which edges have to be enhanced or kept and which ones have to be cancelled. Since there are no analytical solutions for solving the nonlinear diffusion equation, one needs to approximate the solution by discretizing the PDE. The easiest option is to use *explicit schemes*, since these schemes are straightforward to implement. However, they are computationally intense, since many iterations are needed to reach a desired scale level due to stability problems. Another alternative is using *semi-implicit* schemes that can be solved efficiently using Additive Operator Splitting (AOS) schemes [21]. AOS schemes are absolutely stable for any step size and can be used for creating nonlinear scale spaces for feature detection and description problems [7]. However, AOS schemes require solving large systems of linear equations at each time step.

## 2.2 Fast Explicit Diffusion

FED combines the advantages of explicit and semi-implicit schemes while avoiding their shortcomings. FED schemes are motivated from a decomposition of box filters in terms of explicit schemes [7, 8]. Iterated box filters approximate Gaussian kernels with good quality and are easy to implement. The main idea is to perform  $M$  cycles of  $n$  explicit diffusion steps with varying step sizes  $\tau_j$  that originate from the factorization of the box filter:

$$\tau_j = \frac{\tau_{\max}}{2 \cos^2\left(\pi \frac{2j+1}{4n+2}\right)}, \quad (4)$$

where  $\tau_{\max}$  is the maximal step size that does not violate the stability condition of the explicit scheme. The corresponding stopping time  $\theta_n$  of one FED cycle is obtained as:

$$\theta_n = \sum_{j=0}^{n-1} \tau_j = \tau_{\max} \frac{n^2 + n}{3}. \quad (5)$$

Some of the step sizes  $\tau_j$  from Eq. 4 may violate stability conditions. However, due to the similarities between FED and box filtering (always stable), we obtain also a stable

scheme at the end of a FED cycle. The discretization of Eq. 1 using an explicit scheme can be expressed in vector-matrix notation as:

$$\frac{L^{i+1} - L^i}{\tau} = A(L^i)L^i, \quad (6)$$

where  $A(L_i)$  is a matrix that encodes the conductivities for the image and  $\tau$  is a constant time step size such that  $\tau < \tau_{max}$  in order to respect stability conditions. For more details about how to build the matrix  $A(L_i)$  we recommend the reader to check the seminal work of Weickert *et al.* [21]. In the explicit scheme, the solution  $L^{i+1}$  is computed in a direct way from the solution at the previous evolution level  $L^i$  and image conductivities  $A(L^i)$ :

$$L^{i+1} = (I + \tau A(L^i))L^i, \quad (7)$$

where  $I$  is the identity matrix. Considering the a priori estimate  $L^{i+1,0} = L^i$ , a FED cycle with  $n$  variable step sizes  $\tau_j$  is obtained as:

$$L^{i+1,j+1} = (I + \tau_j A(L^i))L^{i+1,j}, \quad j = 0, \dots, n-1. \quad (8)$$

It is important to note here that the nonlinearities from the matrix  $A(L^i)$  are kept constant during the whole FED cycle. Once a FED cycle is done, we compute the new values of the matrix  $A(L^i)$ .

### 3 Accelerated KAZE Features

In this section we describe our novel feature detection and description method. We use FED schemes for building a nonlinear scale space considering anisotropic diffusion. To speed-up the construction of the nonlinear scale space, we embed the FED scheme into a fine to coarse pyramidal framework. The pyramidal strategy and FED schemes allow for fast nonlinear scale space construction, suitable for robust feature detection and description.

#### 3.1 Building a Nonlinear Scale Space with Fast Explicit Diffusion

Firstly, we need to define a set of evolution times from which we can build the nonlinear scale space. The scale space is discretized in a series of  $O$  octaves and  $S$  sub-levels. The set of octaves and sub-levels are identified by a discrete octave index  $o$  and a sub-level one  $s$ . The octave and the sub-level indexes are mapped to their corresponding scale  $\sigma$  (pixels) through the following formula:

$$\sigma_i(o, s) = 2^{o+s/S}, \quad o \in [0 \dots O-1], \quad s \in [0 \dots S-1], \quad i \in [0 \dots M], \quad (9)$$

where  $M$  is the total number of filtered images. Now, we need to convert the set of discrete scale levels in pixel units  $\sigma_i$  to time units, since nonlinear diffusion filtering operates in time units. We use the mapping  $\sigma_i \rightarrow t_i$  described in [2] to convert from pixel units to time units:

$$t_i = \frac{1}{2} \sigma_i^2, \quad i = \{0 \dots M\}. \quad (10)$$

Additionally, the input image can be convolved with a Gaussian of standard deviation  $\sigma_0$  to reduce noise and possible artefacts. From the smoothed input image we compute the contrast factor  $\lambda$  in an automatic way as the 70% percentile of the gradient histogram.

---

**Algorithm 1** Pyramidal FED approach for nonlinear diffusion filtering
 

---

**Input** Image  $L^0$ , contrast parameter  $\lambda$ ,  $\tau_{max}$  and set of evolution times  $t_i$

**Output** Set of filtered images  $L^i$ ,  $i = 0 \dots M$

**for**  $i = 0 \rightarrow M - 1$  **do**

1. Compute diffusivity matrix  $A(L^i)$
2. Set FED outer cycle time  $T = t_{i+1} - t_i$
3. Compute number of FED inner steps  $n$
4. Compute step sizes  $\tau_j$
5. Set Prior  $L^{i+1,0} = L^i$

$L^{i+1} = \text{FEDCYCLE}(L^{i+1,0}, A(L^i), \tau_j)$

**if**  $o_{i+1} > o_i$  **then**

Downsample  $L^{i+1}$  with mask  $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$   
 $\lambda = \lambda \cdot 0.75$

**end if**

**end for**

---



---

**Algorithm 2** FED Cycle
 

---

**function** FEDCYCLE( $L^{i+1,0}, A(L^i), \tau_j$ )

**for**  $j = 0 \rightarrow n - 1$  **do**

$L^{i+1,j+1} = (I + \tau_j A(L^i)) L^{i+1,j}$

**end for**

Return  $L^{i+1,n}$

**end function**

---

Given the input image and the contrast factor we can start the FED scheme. We use  $M - 1$  outer FED cycles and for each cycle we compute the minimum number of inner steps  $n$ . In the case of 2D images, the maximal step size that does not violate stability conditions is  $\tau_{max} = 0.25$ , considering a grid size of 1 pixel for the image derivatives. We consider that each FED outer cycle covers a cycle time  $T = t_{i+1} - t_i$ . The FED cycle time  $\theta_n$  as defined in Eq. 5 covers only a discrete set of values. In order to allow arbitrary cycle times  $T$ , we just need to compute the minimum cycle length  $n$  with  $\theta_n \geq T$  and then multiply the time steps sizes  $\tau_j$  by the factor  $q = T/\theta_n$ .

In order to speed-up the computation of the nonlinear scale space, we embed the FED scheme into a fine to coarse pyramidal approach. Notice that this approach is different than the *Cascadic* FED approach proposed in [7, 8]. In these works, FED schemes were embedded into a coarse to fine pyramidal decomposition. This is because in applications such as image denoising, inpainting or variational optical flow, one is normally interested in the evolution steady state  $t \rightarrow \infty$ . In order to reach the steady state as fast as possible, Cascadic FED propagates the solution from coarse to fine levels. However, in our case we are interested in obtaining a set of filtered images to build a nonlinear scale space from which we can detect and describe features.

Once we reach the last sublevel in each octave, we downsample the image by a factor of 2 using the smoothing mask  $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$  and use that downsampled image as the starting image for the next FED cycle in the next octave. After downsampling the image, we need to modify also the contrast parameter  $\lambda$ . The smoothing mask reduces the contrast of an ideal step edge by 25%, and therefore the contrast parameter needs to be multiplied by 0.75. For a better understanding, we summarize the pyramidal FED approach for nonlinear diffusion filtering in Alg. 1. In the same way, the FED inner cycle iterations are described in Alg. 2.

### 3.2 Feature Detection

We compute the determinant of the Hessian for each of the filtered images  $L^i$  in the nonlinear scale space. The set of differential multiscale operators are normalized with respect to scale,

using a normalized scale factor that takes into account the octave of each particular image in the nonlinear scale space, i.e.  $\sigma_{i,norm} = \sigma_i/2^i$ , and

$$L_{Hessian}^i = \sigma_{i,norm}^2 (L_{xx}^i L_{yy}^i - L_{xy}^i L_{xy}^i). \quad (11)$$

For computing the second order derivatives we make use of concatenated Scharr filters with step size  $\sigma_{i,norm}$ . Scharr filters are known to approximate rotation invariance better than other filters or central differences differentiation [20].

We firstly search for maxima of the detector response in spatial location. At each evolution level  $i$ , we check that the detector response is higher than a pre-defined threshold and it is a maxima in a window of  $3 \times 3$  pixels. This is done to quickly discard non-maxima responses. Then, for each of the potential maxima, we check that the response is a maxima with respect to other keypoints from level  $i + 1$  and  $i - 1$ , respectively directly above and directly below in a window of size  $\sigma_i \times \sigma_i$  pixels. Finally, the 2D position of the keypoint is estimated with sub-pixel accuracy by fitting a 2D quadratic function to the determinant of the Hessian response in a  $3 \times 3$  pixels neighbourhood and finding its maximum.

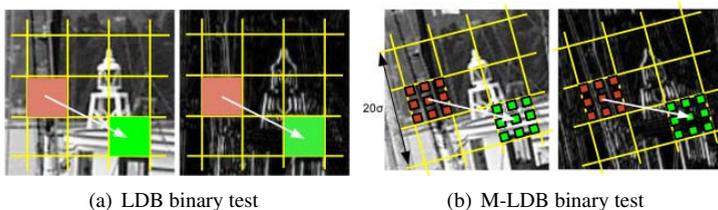


Figure 1: LDB [22] and proposed M-LDB binary tests between grid divisions around a keypoint, shown for the intensity and the gradients in  $x$ . M-LDB includes rotation and sub-sampling that depends on the scale.

### 3.3 Feature Description

Binary descriptors (such as those used in BRIEF, ORB and BRISK) have enjoyed widespread use lately since they can be computed and matched very efficiently. Despite this, only recently insight has been gained into how these binary features actually work. Ziegler *et al.* [23] showed that instead of estimating the gradients of an image as had been previously proposed, these features are a locality sensitive hashing (LSH) of the Kendall’s tau metric. They also introduced the LUCID descriptor, an ordering of the pixels of the patch obtained using a stable sorting algorithm. Computing the Kendall’s tau distance is however an  $O(n \log n)$  operation so they propose to use the Hamming distance over the rankings. The results of LUCID are slightly worse than those of BRIEF, indicating that the dimensionality reduction performed by the hashing does improve performance.

We propose a *Modified-Local Difference Binary* (M-LDB) that exploits gradient and intensity information from the nonlinear scale space. The LDB descriptor was introduced in [22] and follows the same principle as BRIEF, but using binary tests between the average of areas instead of single pixels for additional robustness. In addition to the intensity values, the mean of the horizontal and vertical derivatives in the areas being compared is used, resulting in 3 bits per comparison. LDB proposes using various grids of finer steps, dividing the patch in  $2 \times 2$ ,  $3 \times 3$ ,  $4 \times 4$ , etc. grids, as shown in Fig. 1(a). The averages of

those subdivisions are very fast to compute using integral images if the descriptor is upright (not rotation invariant) as in [22]. However, when considering the rotation of the keypoints integral images can not be used, and visiting all points in a rotated subdivision can be relatively expensive in computation time. Rotation invariance is obtained by estimating the main orientation of the keypoint as in KAZE, and the grid of LDB rotated accordingly. Instead of using the average of all pixels inside each subdivision of the grid, we subsample the grids in steps that are a function of the scale  $\sigma$  of the feature. This approximation of the average performs well in our experiments. The scale-dependent sampling in turn makes the descriptor robust to changes in scale. This process is depicted in Fig. 1(b). M-LDB uses the derivatives computed in the feature detection step, reducing the number of operations required to construct the descriptor.

Given that M-LDB computes an approximation of the average of the same areas in the intensity and gradient images, the Boolean values that result from the comparisons are not independent of each other. Reducing the size of the descriptor by choosing a random subset of the bits [22] or with a more elaborated method such as those used in [6, 16] is expected to improve the results, or at the very least reduce the computational load without decreasing performance.

## 4 Experimental Results

In this section we present experimental results for the combined detection and description performance considering the standard Oxford benchmark [13] with additional experiments under synthetic rotation changes and Gaussian noise. For the latter case we use the *Iguazu* dataset introduced in [2].

We use the VLBenchmarks evaluation from [10] to evaluate the detector repeatability in the Oxford dataset and the synthetic experiments. The VLBenchmark reimplements the protocol introduced in [13] for local detectors evaluation. The detector repeatability measures to what extent do detected regions overlap exactly the same scene region by comparing detected features in two images of the same scene. We consider a correspondence between two regions when the overlap error is smaller than 40%. In addition, we also impose that the error in relative point location for two corresponding regions has to be less than 2.5 pixels, i.e.  $\|x_a - H(x_b)\| < 2.5$ , where  $H$  is the true homography between the images. We compare the performance of A-KAZE with respect to BRISK, ORB, SURF, SIFT and KAZE. For BRISK, ORB, SURF and SIFT we used their OpenCV implementations, while for KAZE we use the original library which is also OpenCV-based<sup>1</sup>.

Fig. 2 depicts the detector repeatability scores for some sequences from the Oxford dataset plus synthetic rotation and Gaussian noise transformations. For the synthetic changes in rotation we consider the first image from the *Graffiti* dataset. We obtained superior detection results with A-KAZE showing excellent performance in all the experiments. KAZE repeatability scores are close to the ones obtained by A-KAZE, but the detector is several orders of magnitude slower. KAZE performs nonlinear diffusion filtering and feature detection always on the same image resolution. On the other hand, A-KAZE downsamples the image size at each new scale speeding-up the nonlinear scale space construction and detection steps. The reason why downsampling does not have a negative impact in the detector performance, is because the nonlinear diffusion process mainly acts within a fairly small vicinity around

<sup>1</sup>Available from <http://www.robosafe.com/personal/pablo.alcantarilla/kaze.html>

each pixel. In addition, another reason why A-KAZE obtains superior results is due to the fact that FED schemes are more accurate than AOS schemes, and therefore the nonlinear diffusion equation is discretized with less error in the FED case.

Regarding fast feature detectors such as ORB and BRISK we found that they undergo a significant drop in performance when adding the pixel location error restriction, especially in the case of BRISK. Note, that our evaluation differs from the one presented in [11] where only overlap area error between features was considered. In addition, the ORB detector does not perform non-maxima suppression between the scales, resulting in duplicate detections within the different pyramid levels in the scale space. We think that one of the reasons why ORB and BRISK have lower detection performance is due to the use of variants of FAST corner detector. As found out in the evaluation of interest points presented in [1], the behaviour of the FAST detector is unreliable for solving image matching problems under large image transformations. For these scenarios, detectors based on the Hessian determinant or Difference of Gaussians should be preferred.

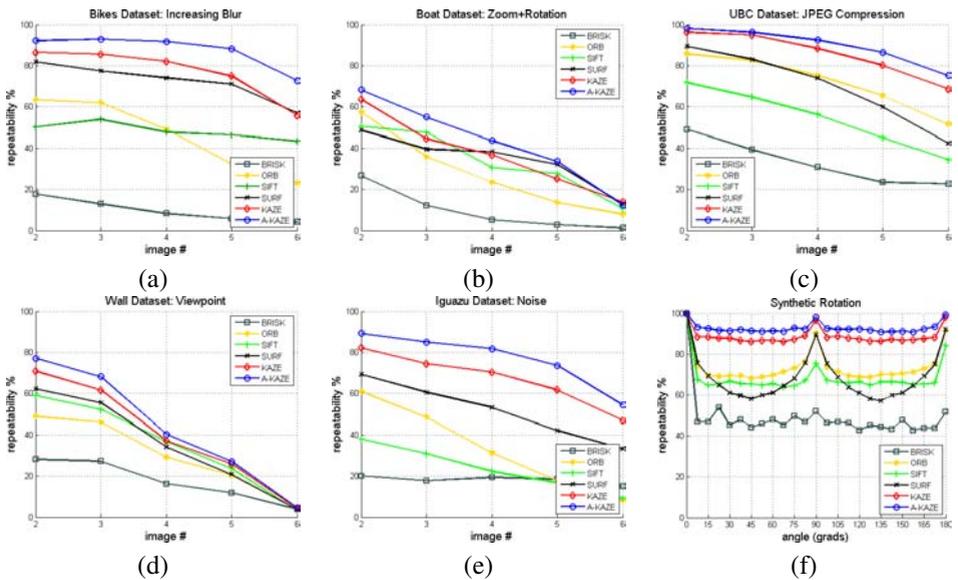


Figure 2: Detector repeatability score for an overlap area error 40% and error in pixel location of 2.5 pixels. (a) Bikes (b) Boat (c) UBC (d) Wall (e) Iguazu (f) Rotation. Best viewed in color.

Table 1 shows average combined detection and description performance results considering the *Matching Score* (MS) and *Recall* (RC) as described in [9]. The matching score  $MS = \# \text{ Correct Matches} / \# \text{ Features}$ , describes the number of initial features that will result in correct matches. The recall  $RC = \# \text{ Correct Matches} / \# \text{ Correspondences}$ , shares some similarities with the MS, although is influenced by the detector repeatability to find correspondences [13]. We consider a nearest neighbor distance ratio matching criteria [12] using a distance ratio value between descriptor distances of 0.8. To determine a correct match we consider the same criteria as done in the detector evaluation, i.e. overlap area error 40% and error in pixel location of 2.5 pixels.

We also tested the impact of selecting various sizes for the descriptor in the perfor-

mance of M-LDB using a random bit selection scheme [22]. As can be observed in Table 1, best MS and RC ratios were obtained by A-KAZE (486 bits) and KAZE in most cases, except for the *Boat* dataset in which SIFT exhibits better recall. Interestingly, A-KAZE with 256 bits obtains very similar performance compared to the full A-KAZE descriptor (486 bits). The light-weight A-KAZE with 64 bits descriptor even outperforms BRISK, ORB, SURF and SIFT in most of the scenarios. Fig. 3(a) shows the percentage

Features	Size	Bikes		Boat		UBC		Trees		Rot.	
		MS	RC	MS	RC	CM	RC	PR	RC	MS	RC
SIFT	128 Bytes	8	69	15	<b>62</b>	33	53	4	29	49	74
SURF	64 Floats	27	68	11	52	53	72	7	23	43	70
ORB	256 Bits	25	67	7	17	59	78	8	34	47	82
BRISK	512 Bits	3	55	3	37	17	51	3	22	15	69
KAZE	64 Floats	37	81	<b>16</b>	54	74	85	<b>18</b>	<b>53</b>	<b>67</b>	85
A-KAZE	64 Bits	42	76	11	33	71	77	12	32	60	86
A-KAZE	256 Bits	46	85	15	44	78	85	16	45	63	91
A-KAZE	486 Bits	<b>47</b>	<b>87</b>	<b>16</b>	47	<b>79</b>	<b>86</b>	17	48	64	<b>92</b>

Table 1: Combined detector and descriptor evaluation results for the *Bikes*, *Boat*, *UBC*, *Trees* and synthetic *Rotation* sequences. The highest MS and RC scores are in bold.

$Precision = \# \text{ Correct Matches} / \# \text{ Putative Matches}$  as a function of the number of bits in the descriptor for the *Trees* dataset. For sizes of 92 bits and above, the performance is within a few percentage points of the full descriptor, and plateaus after 128 bits for small image transformations. Fig. 3(b) depicts precision versus the number of channels considered in the M-LDB descriptor for two image pairs from the *Wall* dataset. There is a considerable improvement in precision when considering the three channels in the descriptor (intensity and  $x$  and  $y$  derivatives) against using only the intensity.

Finally, Fig.3(c) depicts a timing evaluation of the combined detection and description considering 1000 features extracted from the fist image of the Graffiti dataset. This image has a resolution of  $800 \times 640$  pixels. All timing results were obtained with an Intel Core i7-3770 CPU. We consider the OpenCV implementations of BRISK, ORB, SIFT and SURF since these implementations are highly optimized in terms of speed. In addition, we also show timing results for the original SURF library and SIFT from the VLFeat library [18]. Similarly to the original KAZE code, our implementation is also based in OpenCV. While A-KAZE is more expensive to compute than BRISK and ORB, it is faster than SURF, SIFT and KAZE. More specifically, A-KAZE is several orders of magnitude faster than KAZE while providing similar or even better performance in some scenarios. In addition, due to the use of binary descriptors the matching step can be computed very efficiently using the Hamming distance. In the supplementary material, we compare A-KAZE against BRISK and ORB in a tracking by detection scenario considering a high frame rate object tracking sequence from [24].

## 5 Conclusions and Future Work

In this paper we have introduced FED schemes for feature detection and description problems. By means of FED schemes embedded in a pyramidal approach we can reduce dramatically the computational time spent on building the nonlinear scale space and feature

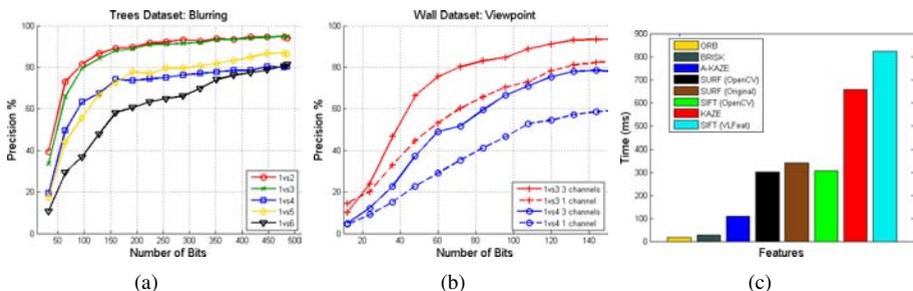


Figure 3: Descriptor and timing evaluation. (a) Precision vs # Bits (b) Precision vs # Channels (c) Timing evaluation for the joint method (detection and description) considering 1000 features from the first image of the Graffiti dataset. Best viewed in color.

detection. In addition, we also introduced a novel modification of the LDB descriptor that is highly efficient and invariant to rotation and scale changes. Our A-KAZE features exhibit excellent performance in detection and description, while having low computational and descriptors storage requirements.

As future work we are interested in developing a GPGPU implementation of A-KAZE since FED schemes are very simple to implement and easily parallelizable. We believe that FED schemes will become a very popular tool in computer vision and image processing applications. FED potential applications are not limited to feature detection and description, and also include many different variational problems [4, 17]. The code of the A-KAZE features is implemented in C++ using some functionalities from the OpenCV library. An open source implementation can be downloaded from: [www.robSAFE.com/personal/pablo.alcantarilla/kaze.html](http://www.robSAFE.com/personal/pablo.alcantarilla/kaze.html).

## Acknowledgments.

The authors would like to thank Andrew J. Davison and Frank Dellaert for useful discussions. Most of Jesus Nuevo’s contributions were made while he was a member of the CI2CV group at CSIRO, Brisbane, Australia.

## References

- [1] H. Aanæs, A.L. Dahl, and K.S. Pedersen. Interesting interest points - a comparative study of interest point performance on a unique data set. *Intl. J. of Computer Vision*, 97(1):18–35, 2012.
- [2] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *Eur. Conf. on Computer Vision (ECCV)*, pages 214–227, 2012.
- [3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [4] R. Ben-Ari and G. Raveh. Variational depth from defocus in real-time. In *Computer Vision Workshops, IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 522–529, 2011.

- [5] M. Calonder, V. Lepetit, M. Özuysal, T. Trzinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Machine Intell.*, 34(7):1281–1298, 2011.
- [6] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2887–2894, 2012.
- [7] S. Grewenig, J. Weickert, C. Schroers, and A. Bruhn. Cyclic Schemes for PDE-Based Image Analysis. Technical Report 327, Department of Mathematics, Saarland University, Saarbrücken, Germany, March 2013.
- [8] S. Grewenig, J. Weickert, and A. Bruhn. From box filtering to fast explicit diffusion. In *Proceedings of the DAGM Symposium on Pattern Recognition*, pages 533–542, 2010.
- [9] J. Heinly, E. Dunn, and J. M. Frahm. Comparative evaluation of binary features. In *Eur. Conf. on Computer Vision (ECCV)*, pages 759–773, 2012.
- [10] K. Lenc, V. Gulshan, and A. Vedaldi. VLBenchmarks. <http://www.vlfeat.org/benchmarks/>, 2012.
- [11] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 2548–2555, 2011.
- [12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004.
- [13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Intl. J. of Computer Vision*, 65(1-2):43–72, 2005.
- [14] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(7):1651–1686, 1990.
- [15] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Eur. Conf. on Computer Vision (ECCV)*, Graz, Austria, 2006.
- [16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 2564–2571, 2011.
- [17] A. Schmidt-Richberg, J. Ehrhardt, R. Werner, and H. Handels. Fast explicit diffusion for registration with direction-dependent regularization. In *Intl. Workshop on Biomedical Image Registration (WBIR)*, pages 220–228, 2012.
- [18] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [19] S. Wang, H. You, and K. Fu. BFSIFT: A novel method to find feature matches for SAR image registration. *IEEE Geosci. Remote Sensing Lett.*, 9(4):649–653, 2012.
- [20] J. Weickert and H. Schar. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, 13:103–118, 2002.

- [21] J. Weickert, B. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Processing*, 7(3):398–410, 1998.
- [22] X. Yang and K. T. Cheng. LDB: An ultra-fast feature for scalable augmented reality. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2012.
- [23] A. Ziegler, E. Christiansen, D. Kriegman, and S. Belongie. Locally uniform comparison image descriptor. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [24] K. Zimmermann, T. Svoboda, and J. Matas. Tracking by an optimal sequence of linear predictors. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(4):677–692, 2009.