# A Low Complexity Interest Point Detector

Jie Chen, *Student Member, IEEE*, Ling-Yu Duan, *Member, IEEE*, Feng Gao, Jianfei Cai, *Senior Member, IEEE*, Alex C. Kot, *Fellow, IEEE*, and Tiejun Huang, *Senior Member, IEEE*

***Abstract*—Interest point detection is a fundamental approach to feature extraction in computer vision tasks. To handle the scale invariance, interest points usually work on the scale-space representation of an image. In this letter, we propose a novel block-wise scale-space representation to significantly reduce the computational complexity of an interest point detector. Laplacian of Gaussian (LoG) filtering is applied to implement the block-wise scale-space representation. Extensive comparison experiments have shown the block-wise scale-space representation enables the efficient and effective implementation of an interest point detector in terms of memory and time complexity reduction, as well as promising performance in visual search.**

***Index Terms*—Block-wise scale-space representation, interest point detector, Laplacian of Gaussian, scale-space.**

## I. Introduction

WITH the explosive growth of mobile devices, deploying computer vision algorithms at low computational complexity is important. For example, to reduce query delivery latency, recent works in mobile visual search [1], [2], [3], [4], [5] have attempted to extract and compress visual features directly on a mobile device and transmit compact descriptors instead of images to the remote server. As intense computing may consume considerable power and shorten battery life, the complexity of feature extraction should be moderate.

Local features exhibit superior performance in image recognition and classification. Local feature extraction typically involves detecting interest points and describing the invariant feature of each interest point. Ideally, an interest point detector should be invariant to transformations like scale change, translation and rotation. In this letter, we propose a novel approach to significantly reduce the complexity of an interest point detector in terms of memory and time. In particular, a low complexity interest point detector is a key requirement in the emerging MPEG standard on Compact Descriptors for Visual Search (CDVS)[6].

Manuscript received June 13, 2014; accepted August 15, 2014. Date of publication September 04, 2014; date of current version September 09, 2014. This work was supported by the National Natural Science Foundation of China under Contracts 61271311, 61121002, 61390515, and 61210005. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Alex Dimakis.

J. Chen, L.-Y. Duan, F. Gao, and T. Huang are with the Institute of Digital Media, the School of EE & CS, Peking University, Beijing 100871, China (e-mail: cjie@pku.edu.cn, lingyu@pku.edu.cn, gaof@pku.edu.cn, tjhuang@pku.edu.cn). Ling-Yu Duan is the corresponding author.

J. Cai is with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: ASJFCai@ntu.edu.sg).

A. C. Kot is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: eackot@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2014.2354237

To handle the scale invariance, an image is extended to a scale-space, which is represented by a family of smoothed images, parameterized by a series of smoothing kernels of different scales. The scale-space representation involves two basic problems: how to construct a scale-space and how to identify interest points within a scale-space. In [7], [8], [9], it has been proved that Gaussian function is the unique smoothing kernel for constructing a scale-space, satisfying the scale-space axioms [9] including linearity, shift invariance, semi-group structure, non-creation of local extrema, scale invariance and rotational invariance. To identify interest points within a scale-space, Crowley and Parker [10] proposed to identify peaks and ridges in the low-pass transform of an image (which is a kind of scale-space derivative). Lindeberg proposed automatic scale selection based on local extrema (i.e., maximum/minimum) over the normalized scale-space derivatives through different scales [9], [11], [12].

In practice, a scale-space is represented as an image pyramid in which an image is successively filtered by a family of smoothing kernels at increasing scale factors. Meanwhile, the normalized derivatives of each scale in an image pyramid are generated, where extrema detection is performed by searching for local extrema, to identify interest points.

However, such an image pyramid based scale-space representation requires huge amount of memory to store the smoothed images and normalized derivatives, and incurs heavy convolution operations for generating smoothed images and differential operations for normalized derivatives. Lowe [13] proposed a Difference of Gaussian (DoG) filter to approximate a Laplacian of Gaussian (LoG), which reduces the differential operations of Laplacian to simple subtractions of Gaussian scales. Bay [14] used box filters and integral images to approximate the Determinant of Hessian of Gaussian scales, thereby saving the convolution operations and differential operations. However, neither of two methods has addressed the issue of heavy memory cost in constructing a scale-space representation. Moreover, further reducing the filtering time cost in a pyramid based scale-space is meaningful [6] to improve the efficiency of scale-space construction.

To address the memory complexity issue, we propose a block-wise scale-space representation. By decomposing an image into blocks, we construct a block-wise scale-space, and perform extrema detection within a block of scale-space, thereby significantly reducing the memory cost of filters and buffers (say, $< 1\text{M}$ bytes). Meanwhile, the block-wise scale-space enables the frequency domain filtering mechanism, thereby reducing time cost by $2 \sim 3$ times. In particular, the implemented Block based Frequency Domain Laplacian of Gaussian (BFLoG) detector has been adopted by the emerging MPEG CDVS standard [15].

The rest of this letter is organized as follows: Section II presents the problem of a block-wise scale-space. Section III describes the block-wise scale-space and its low complexity characteristics. Extensive experiments are shown in Section IV. Finally, we conclude this letter in Section V.

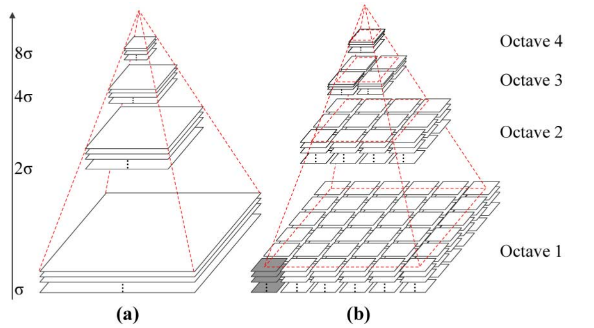


Fig. 1. The proposed pyramid of block-wise scale-space (b) to approximate that of image-wise scale-space (a).

## II. Problem Statement

### A. Brief Review of the Scale-Space

The scale-space of an image $\mathrm{I}(x, y)$ is defined as a function $\mathrm{L}(x, y, \sigma)$ in the continuous domain [13], which convolves the input image with a series of Gaussian functions $\mathrm{G}(x, y, \sigma)$ of different scale factors:

$$\mathrm{L}(x, y, \sigma) = \mathrm{G}(x, y, \sigma) * \mathrm{I}(x, y) \tag{1}$$

where $*$ denotes a convolution operation, and

$$\mathrm{G}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

As concluded in [16], the local extrema of the Laplacian of Gaussian (LoG) with scale normalization produce most of the stable interest points, with the formulation as follow:

$$\mathrm{LoG}(x, y, \sigma) = \sigma^2 \nabla^2 \mathrm{L}(x, y, \sigma) \tag{3}$$

where $\nabla^2$ denotes a Laplacian operator.

As a discrete representation of the scale-space, a number of scale sampled images are generated w.r.t. scale factors $\sigma$. With the increase of scale factor, the enlarged Gaussian Kernels bring about more complex convolution operations. To reduce time complexity, rather than memory complexity, a scale-space is usually represented as an image pyramid as illustrated in Fig. 1(a), where the sampled images are grouped into $Q$ octaves $\{O_q | q = 1, \ldots, Q\}$. Each octave consists of $S$ smoothed images with exponentially sampled scale factor $\sigma_k = 2^{\frac{k}{S}} \sigma_0$, $k = 1, \ldots, S$. Once an octave is constructed, the smoothed image that has twice the initial value $\sigma_0$ is downsampled as the input image for the next octave. Such an image pyramid saves the convolution operations a lot. On the one hand, within each octave, scale images can be generated with a cascade of reduced sized filters. On the other hand, as the scale factor reaches up to the twice of the initial value $\sigma_0$, the smoothed image can be downsampled with a factor of 2 due to redundant pixel information. Accordingly, the kernel size is halved.

### B. Problem Formulation

We aim to decompose a scale-space into block-wise representation and perform interest point detection for each block independently. Such block-wise representation may degenerate compared to the original scale-space, which would impact the performance (say, image matching) of local features. We derive the block-wise scale-space representation by resolving the problem of minimizing the distortion of scale-space, subject to the constraints of memory and time cost.

Given an image I, the objective is to minimize the distortion D between the original scale-space L and the block-wise scale-space $\overline{\mathrm{L}}$, by optimizing the block decomposing configuration $\mathbf{B}$, as follows

$$\min_{\mathbf{B}} \ \mathrm{D}(\mathrm{L}, \overline{\mathrm{L}}(\mathbf{B})) \tag{4}$$

For each octave $O_q$, we denote the block decomposition configuration $\mathbf{B}$ as follows

$$\mathbf{B} = \{B_{q,k}\}_{k=1}^{K_q} = \{(x_{q,k}, y_{q,k}, w_{q,k}, h_{q,k})\}_{k=1}^{K_q} \tag{5}$$

where $x_{q,k}, y_{q,k}, w_{q,k}, h_{q,k}$ denote the location coordinate, width, height of each block, respectively; $K_q$ the block number within octave $O_q$, and the union of $K_q$ decomposed blocks shall cover the original octave sufficiently.

*Distortion Measure:* The distortion $\mathrm{D}(\mathrm{L}, \overline{\mathrm{L}})$ can be directly measured by Mean Square Error (MSE) between the original scale-space L and the block-wise scale-space $\overline{\mathrm{L}}$. However, the MSE distortion cannot explicitly reflect the performance drop in vision tasks like visual search, scene classification, etc., from the degenerated interest point detection in the distorted scale-space. In this work, we propose a novel distortion measure in terms of image matching precision (MP), which is a crucial criteria for evaluating visual search performance. MP can be defined as a True Positive Rate (TPR) at a target False Positive Rate (FPR) over an image collection of image matching pairs and non-matching pairs. Hence, the distortion $\mathrm{D}(\mathrm{L}, \overline{\mathrm{L}})$ in (4) is reduced to the degradation of image matching precision MP between the original scale-space L and the block-wise scale-space $\overline{\mathrm{L}}$ with an optimal block configuration $\{B_{q,k}\}_{k=1}^{K_q}$:

$$\mathrm{D}(\mathrm{L}, \overline{\mathrm{L}}(\mathbf{B})) = MP_{\mathrm{L}} - MP_{\overline{\mathrm{L}}} \tag{6}$$

*Constraints:* The block-wise scale-space allows for the block-level interest point detection to be independent of each other, which contributes to the significant reduction of memory. A simple way is to uniformly partition the image into blocks or stripes (a few consecutive lines of pixels). A block based partition can greatly minimize the memory cost, while a stripe based partition suits for the raster scanning of image capturing but save less memory cost. In terms of the constraint to minimize the performance distortion $\mathrm{D}(\mathrm{L}, \overline{\mathrm{L}})$, the size of the largest block determines the upper limit of memory use. Let $\mathrm{Size}(B_{q,k}) = w_{q,k} \times h_{q,k}$ denote the size of each block $B_{q,k}$. The block decomposition optimization in (5) is subject to the maximum of $\mathrm{Size}(B_{q,k})$. In other words, the optimal block configuration $\{B_{q,k}\}_{k=1}^{K_q}$ is subject to the memory use in addition to the performance distortion.

## III. Block-Wise Scale-Space

### A. Building up the Block-Wise Scale-Space

To construct a block-wise scale-space, we partition the original scale-space into $K_q$ overlapping blocks. As interest point detection is performed for each separate block, proper overlapping is to minimize the precision loss of scale-space representation at the boundary of each block when performing Gaussian or Laplacian of Gaussian convolution. Ideally, minimizing the distortion in (4) needs to walk through a huge parameter space $\{(x_q^k, y_q^k, w_q^k, h_q^k)\}$ to search for the best configuration $\{B_{q,k}\}_{k=1}^{K_q}$, with respect to all blocks in all octaves in an image pyramid. Clearly, this is intractable.

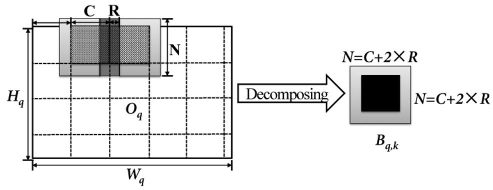


Fig. 2. Decomposing an image into overlapped blocks.

The proposed block-wise scale-space is described as below: Starting from the top left corner, each octave input image $O_q$ is decomposed into $K_q$ square $R$-pixel overlapped blocks $B_{q,k}$. See Fig. 2, the width (or height) of each block is $N = C + 2R$. $\mathrm{Size}(B_{q,k}) = N^2$. The objective in (4) can be reduced to:

$$\min_{N,R} \mathrm{D}(\mathrm{L}, \overline{\mathrm{L}}(\mathbf{B}(N,R)))$$
$$s.t. \quad 0 < N \leq N_{\max} \quad and \quad 0 \leq R < \frac{N}{2} \tag{7}$$

where $N_{\max}$ determines the maximum width (or height) of each block to reflect the memory constraint. Thus, the huge parameter space is reduced to a tractable space of two parameters $N$ and $R$. For the nature of convolution, both parameters impact the performance of interest points detection in the block-wise scale-space representation. This has been validated in experiments in Section IV. Since both $N$ and $R$ are integer and limited with the width or height of an image, the range of $N$ and $R$ are enumerable. Therefore, to solve (7), a straightforward way is to enumerate the values of $N$ and $R$, and determines the optimal configuration by choosing the minimum matching performance distortion.

With an optimal configuration $B_{q,k}$, the scale-space of each block $k$ for each octave $O_q$ involves $S$ Gaussian representation $B_{q,k}^{G_a}, a \in \{1 \ldots S\}$, and $S + 2$ LoG representation $B_{q,k}^{LoG_b}, b \in \{0 \ldots S + 1\}$. Two additional layers of LoG representation is meant to support the neighborhood comparison of LoG response values in extrema detection along the scale direction. To locate interest points, we perform extrema detection in the scale-space of each block. Following [13], we generate the local extrema by comparing the response value of a point to the neighbor $U(x, y, \sigma)$ (say, $3 \times 3 \times 3$). Due to the overlap $R$, the correct comparison of response values at the block boundary can be secured. Finally, we generate the set of interest points $\{(\hat{x}, \hat{y}, \hat{\sigma})\}$ which is the union of interest points obtained by examining the extrema of $B_{q,k}^{LoG}$ for each block in each octave.

$$\{(\hat{x}, \hat{y}, \hat{\sigma})\} = \bigcup_{q=1..Q} \bigcup_{k=1..K_q} (\arg \max_{(x,y,\sigma)} (B_{q,k}^{LoG}(x, y, \sigma)),$$
$$(\arg \min_{(x,y,\sigma)} (B_{q,k}^{LoG}(x, y, \sigma)) \quad s.t. \quad (x, y, \sigma) \in U(x, y, \sigma) \tag{8}$$

### B. Complexity Analysis of the Block-Wise Scale-Space

The block-wise scale-space has clear potentials of low memory complexity. As interest point detection works on each block independently, the actual memory cost equals to the memory cost of an individual block. Referring to (4), the optimal configuration $\mathbf{B}$ is subject to the memory use.

Below we discuss time complexity. Convolution operations are the most time consuming in constructing a scale-space. Time complexity $\mathrm{T}(N, R)$ can be measured in terms of block number and time cost of a series of filtering operations of a block as follows:

$$\mathrm{T}(N, R) = \sum_{q=1}^{Q} f_{construction}(N) * \left\lceil \frac{W_q}{N - 2R} \right\rceil * \left\lceil \frac{H_q}{N - 2R} \right\rceil \tag{9}$$

where the number of blocks in each octave is $\lceil \frac{W_q}{N-2R} \rceil * \lceil \frac{H_q}{N-2R} \rceil$, $W_q$ and $H_q$ are the image width and height of the octave input image respectively. $f_{construction}$ denotes the filtering time cost of each block. Below we discuss how to speed up the construction of a block-wise scale-space.

(1) **Frequency domain convolution.** The block-wise process allows for very efficient frequency domain filtering. In the spatial domain, each point of an input image is smoothed by its neighbor region where the region radius increases with a scale factor. The equivalent of convolution in frequency domain is independent of the scale factor, where the convolution involves only one dot product operation per point, a Discrete Fourier Transform (DFT), and a Inverse Discrete Fourier Transform (IDFT). As each block is of a uniform size, convolution filters can be pre-computed by different scale factors. In addition, we employ the Cooley-Tukey Fast Fourier Transform (FFT) algorithm with the complexity:

$$O_{FFT}(N) = 2\bar{N}^2 \log_2 \bar{N}^2 - 6\bar{N}^2 + 8\bar{N} \tag{10}$$

Here, $\bar{N}$ is the closest number of power of 2 equal or larger than $N$. A larger kernel size means even faster FFT filtering compared to the spatial domain convolution.

(2) **Paralleled implementation.** The independent interest point detection of each block offers the engineering potential of paralleled implementation to speed up the process.

## IV. Experiments

*Datasets and Evaluation Protocols:* We evaluate the performance of BFLoG over the MPEG CDVS benchmark datasets [22]. There are in total 30,256 images involving a variety of categories like Graphics, Paintings, Video Frames, Landmarks, and Common Objects, which are well annotated in terms of matching pairs (10,155) vs. non-matching pairs (112,175), query images (8,313) versus reference images (18,840). Moreover, a FLICKER1M dataset containing 1 million images is used as distractors in retrieval experiments.

All images are converted to grayscale images. If at least one of the dimensions of the original image is greater than 640 pixels then the original image shall be spatially resampled, maintaining the aspect ratio, so that the largest of the vertical and horizontal image dimensions is equal to 640 pixels.

The retrieval performance is measured by mean Average Precision (mAP). The pairwise matching performance is measured by a True Positive Rate (TPR) at a target False Positive Rate (FPR), where 1% FPR is setup in our experiments.

*Baselines:* We perform comparisons over six baselines: (1) $BFLoG + SIFT$: combines the BFLoG detector and the well known SIFT descriptor [13]; (2) $DoG + SIFT$ [13]: DoG is well known for a fast approximation of LoG. (3) *SURF* [14]: a fast detector by applying integral images to image convolutions. (4) *AKAZE* [17]: using a non-linear scale-space construction kernel of higher complexity than a linear scale-space kernel. (5) *ORB* [18]: a very fast detector ignoring the scale invariance. (6) *BRISK* [19]: injecting the scale invariance to the AGAST corner detector [20]. Comparing baselines (1) and (2) is to validate the performance of a block-wise scale-space at much lower memory cost. Other baselines are for extensive comparisons with the state-of-the-art local feature descriptors.

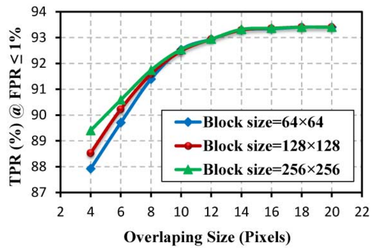


Fig. 3. Matching performance w.r.t. Block overlap size.

TABLE I
COMPARISON OF FILTERING TIME COMPLEXITY IN THE SCALE-SPACE CONSTRUCTION BETWEEN FREQUENCY DOMAIN BFLOG AND SPATIAL DOMAIN DOG (UNIT: MFLOP)

| Octave | | Blocks | Frequency filtering | | | Spatial filtering | Ratio |
|---|---|---|---|---|---|---|---|
| W | H | | FFTs | Filtering | Total | | |
| 640 | 480 | 35 | 113.86 | 2.29 | 116.16 | 259.28 | 45% |
| 320 | 240 | 12 | 39.04 | 0.79 | 39.83 | 55.6 | 72% |
| 160 | 120 | 4 | 13.01 | 0.26 | 13.28 | 13.9 | 95% |
| 80 | 60 | 1 | 3.25 | 0.07 | 3.32 | 3.48 | 95% |
| 40 | 30 | 1 | 3.25 | 0.07 | 3.32 | 0.87 | 382% |
| Total | | 53 | 172.42 | 3.47 | 175.9 | 333.12 | 53% |

*Distortion Analysis:* As shown in Fig. 3, by fixing a block size, increasing block overlap $R$ can improve TPR till $R$ reaches the maximum radius of Gaussian filters (i.e., $R = 16$) and TPR becomes stable. Given a fixed overlap $R < 16$, a larger block means fewer blocks, which yields higher performance due to less accumulated convolution loss at the boundaries. In subsequent experiments, we set overlap $R = 16$.

*Complexity Analysis:* For time cost, we measure the Million FLoating-point OPeration (MFLOP) in constructing a scale-space. For memory cost, we provide a theoretical estimate of memory use of filtering and buffering, etc..

As shown in Fig. 4(b), a bigger block incurs more memory use. Fig. 4(a) does not show any monotone increasing or decreasing patterns w.r.t. the block size. On the one hand, this is due to the nature of DFT that a block width of power of 2 is with lower complexity. On the other hand, too large blocks may result in more "waste" operations from the padding pixels of a block at the boundary of an image, while too small blocks result in more operations at the overlapping regions of more blocks. Thus, an optimal block width $N = 128$ is empirically setup for the subsequent comparison experiments.

(1) Time Cost. Table I compares the time cost of filtering (the major part in constructing a scale-space) between frequency and spatial domain. The frequency domain filtering shows a significant time cost reduction by a ratio of 53% on average.

(2) Memory Cost. Our BFLoG has reduced the footprint of filters and buffers to 956KB, which is much smaller than the image-wise scale-space with 12.9 MB in baseline (2).

*Performance Comparisons:* As shown in Fig. 5, the BFLoG has yielded comparable or slightly better performance than the DoG in pairwise matching and retrieval experiments, at much reduced runtime memory and time cost(See Table II).

Fig. 6 compares the matching TPR of all the baselines. The BFLoG achieves the highest TPR. Table II reports the average number of interest points, the extraction time and the runtime memory cost of baselines. The BFLoG is the second fastest with

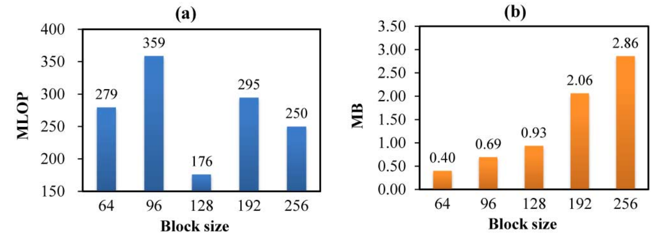


Fig. 4. The impact of different block sizes on time cost (a) and memory use (b).

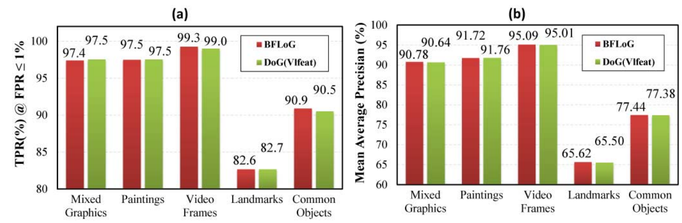


Fig. 5. Performance comparison between the BFLoG and the DoG for pairwise matching (a) and retrieval (b).

TABLE II
RUNTIME MEMORY AND TIME COST COMPARISON OF BASELINE DETECTORS. (TESTED ON A WINDOWS PC WITH INTEL CORE CPU I5 3470@3.2 GHZ). NOTE THAT THE RUNTIME MEMORY COST IS GREATER THAN THE THEORETICAL ESTIMATED MEMORY USE IN TERMS OF FILTERS AND BUFFERS AS REPORTED IN COMPLEXITY ANALYSIS

| | Ave. Number of Interest Points | Time | Memory |
|---|---|---|---|
| BFLoG | 1011.59 | 132ms | 5.3MB |
| SIFT(VlFeat) | 1081.46 | 383ms | 30.2MB |
| SURF | 1164.97 | 225ms | 50.4MB |
| AKAZE | 917.77 | 171ms | 76.4MB |
| ORB | 1330.89 | 22ms | 9.2MB |
| BRISK | 917.03 | 234ms | 24.9MB |

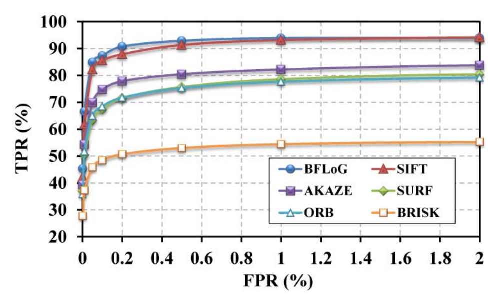


Fig. 6. Performance comparison among different descriptors over MPEG CDVS benchmark datasets.

the lowest runtime memory cost. However, the fastest ORB suffers from the poor performance as shown in Fig. 6.

## V. CONCLUSION

We have proposed a block-wise scale-space representation to minimize the complexity of an interest point detector. Very fast extraction and extremely low memory footprint have been achieved for the well-known LoG detector. The block-level independent behavior of filtering and extrema detection elegantly supports a paralleled process. The MPEG CDVS standard adopted BFLoG has shown great advantages of block-wise scale-space representation towards low complexity visual feature extraction on (mobile) hardware platforms like multi-core CPU, GPU, DSP, or ASIC.

## REFERENCES

[1] R. Ji, L.-Y. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, and W. Gao, "Location discriminative vocabulary coding for mobile landmark," in *IJCV*, 2012, vol. 96, pp. 290–314.

[2] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, "Compact descriptors for visual search," *IEEE Multimedia Mag.*, 2014.

[3] B. Girod, V. Chandrasekhar, D. Chen, N. Cheung, R. Gizeszczuk, Y. Reznik, and R. Vedantham, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, 2011.

[4] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao, "Rate-adaptive compact fissher codes for mobile visual search," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 195–198, 2014.

[5] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor," *CVPR*, 2009.

[6] ISO/IEC JTC1/SC29/WG11/N11531, "Compact descriptors for visual search: Requirements," Jul. 2010. Geneva, Switzerland.

[7] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, vol. 50, no. 5, pp. 363–370, 1984.

[8] L. M. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "Scale and the differential structure of images," *Image Vis. Comput.*, vol. 10, no. 6, pp. 376–388, 1992.

[9] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *J. Appl. Statist.*, vol. 21, no. 2, pp. 224–270, 1994.

[10] J. L. Crowley and A. C. Parker, "A representation for shape based on peaks and ridges in the difference of low-pass transform," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 6, no. 2, pp. 156–170, 1984.

[11] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention," in *IJCV*, 1993, vol. 11, no. 3, pp. 283–318.

[12] T. Lindeberg, "Feature detection with automatic scale selection," in *IJCV*, 1998, vol. 2, pp. 77–116.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *IJCV*, 2004, vol. 60, pp. 91–110.

[14] H. Bay, T. Tuytelaars, and L. Gool, "SURF: Speeded up robust features," in *ECCV*, 2006.

[15] ISO/IEC JTC1/SC29/WG11/m33159, "CDVS CE1: A low complexity detector ALP_BFLoG," Apr. 2014. Valencia, Spain.

[16] K. Mikolajczyk, "Detection of local features invariant to affine transformations," Ph.D. thesis, Inst. Nat. Polytechnique Grenoble, Grenoble, France, 2002.

[17] P. F. Alcantarilla and T. V. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, 2011.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to sift or surf," in *ICCV*, 2011.

[19] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariantscalable keypoints," in *ICCV*, 2011.

[20] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *ECCV*, 2010.

[21] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *ACMMM*, 2010.

[22] ISO/IEC JTC1/SC29/WG11/N12202, "Evaluation framework for compact descriptors for visual search," 2011.