



HAL
open science

An image smoothing operator for fast and accurate scale space approximation

Maxim Karpushin, Giuseppe Valenzise, Frédéric Dufaux

► **To cite this version:**

Maxim Karpushin, Giuseppe Valenzise, Frédéric Dufaux. An image smoothing operator for fast and accurate scale space approximation. IEEE International Conference on Acoustics, Speech, and Signal Processing, Mar 2016, Shanghai, China. hal-01299839

HAL Id: hal-01299839

<https://hal.science/hal-01299839>

Submitted on 8 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN IMAGE SMOOTHING OPERATOR FOR FAST AND ACCURATE SCALE SPACE APPROXIMATION

Maxim Karpushin, Giuseppe Valenzise, Frédéric Dufaux

LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay

ABSTRACT

Gaussian image smoothing is a fundamental operation in the extraction of scale-invariant feature points. Its computation, however, can be too expensive in some resource-constrained scenarios. Alternative solutions such as the box filter can be computed more efficiently, at the cost of a loss in feature repeatability under some conditions. In this paper we propose a fast and accurate image smoothing operator based on integral images. It has the same order of computational complexity as the box filter, but provides much more accurate visual results and improved keypoint repeatability, which is confirmed in a feature detection scenario using SIFT features.

Index Terms— Image smoothing, scale space, box filter, Gaussian filter, SIFT

1. INTRODUCTION

Low pass image filtering (smoothing) is a basic operation in many image processing applications, including image matching through local features. For instance, Gaussian smoothing is used to generate a scale space and detect interesting points in a scale-invariant manner [2]. While this approach can provide very stable keypoints, its computation may not be feasible when power or battery resources are limited. As an alternative, one can use simpler and faster smoothing techniques such as the box filter [6], although this can imply lower feature matching performance under some transformations. The trade-off between computational efficiency and accuracy of the smoothing filter is therefore a key factor in the stability and repeatability of extracted features.

In this paper we introduce a fast and accurate image smoothing filter based on integral images. In terms of computational complexity, the proposed approach is similar to the box filter, as the response is computed in constant time at any image point and any smoothing level. However, our filter provides improved rotational invariance and a better approximation of Gaussian smoothing.

The rest of the paper is organized as follows. In Section 2 we review some useful properties of the Gaussian and the box filter. Section 3 describes the proposed filtering approach in detail. In Section 4 we illustrate the results and performance of the proposed filter, in terms of computation time and rotational invariance, and evaluate its use in a keypoint detection scenario. Finally, Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

Image smoothing is a well-explored research area. Some recent works in it are focused on complexity reduction, e.g. [1]. The focus of this paper is image smoothing complexity in context of salient visual point detection.

2.1. Gaussian smoothing

Bi-dimensional Gaussian filter is one of the most commonly used image smoothing operators. To smooth an input image $f(x, y)$ up to a given smoothing level σ , one would typically compute the following quantity:

$$f_{out}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \iint_{\mathbb{R}^2} f(x, y) e^{-\frac{(x-u)^2+(y-v)^2}{2\sigma^2}} dudv. \quad (1)$$

Thanks to the exponential decay, the integral may be contracted to a reasonably compact support, for example applying the well known rule of 3σ [2]. Moreover, the convolutional kernel is separable, allowing to replace the two-dimensional convolution by two simple ones. The numerical filter, obtained by replacing the integrals by integer summations, has *linear complexity* in function of σ for computing the response in a given spatial point. In other words, discrete version of $f_{out}(x, y, \sigma)$ at a given point (x, y, σ) may be computed in $O(\sigma)$ operations.

We recall in the following two important properties of the Gaussian filter:

Relation to the heat diffusion equation. It is known that the expression (1) is the solution of the differential problem for the heat diffusion equation. The input image becomes the initial condition of this problem, and the amount of injected smoothing σ is related to the diffusion time. This enables to establish a set of important properties, allowing to use this filter to engender a proper scale space [3]. The latter may be understood as a representation of the internal image structure at multiple scales. This kind of representations is predominant in vision problems, such as keypoint detection for image matching [4].

Rotational invariance. The Gaussian convolutional kernel is radially symmetric. This implies perfect invariance of the filter response to in-plane image rotations.

A notable successful application of the Gaussian smoothing motivated by these properties is SIFT image features [2]. Similarly, the Gaussian scale space is employed in the MPEG Compact Descriptors for Visual Search standard [5]. However, due to the linear computational complexity, this filter has been systematically criticized [6, 7, 8, 9], and approximate solutions such as the box filter have become popular.

2.2. Box filter

The box filter response is given by the following expression:

$$f_{out}(x, y, s) = \frac{1}{s^2} \int_{x-\frac{s}{2}}^{x+\frac{s}{2}} \int_{y-\frac{s}{2}}^{y+\frac{s}{2}} f(u, v) dudv \quad (2)$$

The filter thus may be seen as a convolution with a kernel taking a constant value within the rectangular support $\Omega = [-\frac{s}{2}, \frac{s}{2}] \times [-\frac{s}{2}, \frac{s}{2}]$, i.e., taking the average image value in Ω . Here s represents the scale parameter, or the amount of smoothing required on the output. Integral image technique [10] allows to avoid the explicit computation of the integral, providing *constant computational complexity*, i.e., independent of the amount of smoothing s , provided that the integral image has been precomputed. This principle is illustrated on Fig. 1. Due to its simplicity and efficiency, the box

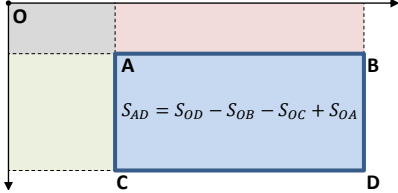


Fig. 1. Integral image principle. For a given function, the integral over any rectangle AD may be computed immediately, if for any point X on the plane the integral over OX is known. The latter ones are precomputed once during the filter initialization stage, and form the integral image (the rectangles are denoted by their diagonals).

filter is largely used in different applications and scenarios. Some complex image filters, such as the guided filter [11] or some bilateral filter variant [12], employ the box filtering. In the vision applications, the box filter is often used to approximate the time-consuming Gaussian scale space in SIFT-like detectors. Thus, [7] proposes to select the keypoint candidates in a *Difference-of-Mean* (DoM) image pyramid built with the box filtering instead of the original *Difference-of-Gaussian* pyramid. In a similar way integral images [8] and their generalizations [9] are combined with SIFT detection strategy. Speeded Up Robust Features (SURF) [6] also make use of the box filtering for the keypoint detection. Center Surrounded Extrema (CenSurE) detector [13] uses slanted integral images to detect the interesting points without computing explicitly a pyramidal representation. BRISK [14] employs the box smoothing to compute the descriptor.

An issue when using the box filter in vision problems comes from its sensitivity to rotations. Due to the sharp corners of the rectangular convolutional kernel, rotating the input image of 45° may drastically change the distribution of interesting points with respect to the structure of the image. For instance, it is known that SURF features suffer from limited rotational invariance, which is partially due to the use of box filtering [6].

3. THE PROPOSED FILTER DESIGN

3.1. Filter kernel definition

The key idea of the proposed approach consists in approximating the Gaussian kernel by a function that can be computed using *image moments*, which are efficiently represented by the integral images.

Let us consider the following function:

$$K(x, y) = A - B(x^2 + y^2), \quad (3)$$

where A and B are constants. It is straightforward to show that the convolution F of the image with this kernel may be decomposed as

follows:

$$\begin{aligned} F(x, y, s) &= \iint_{\Omega} f(u, v)K(x - u, y - v)dudv \\ &= [A - B(x^2 + y^2)] I_1(x, y, s) - BI_{x^2+y^2}(x, y, s) \\ &\quad + 2B [xI_x(x, y, s) + yI_y(x, y, s)] \end{aligned} \quad (4)$$

The integrals $I_{(\cdot)}$ denote *image moments*. Specifically, I_1 is the zero-order moment that is equivalent to the box filter output, I_x and I_y are first-order moments with respect to x and y , and $I_{x^2+y^2}$ is the sum of two second-order moments:

$$I_x(x, y, s) = \iint_{\Omega} uf(u, v)dudv \quad (5)$$

$$I_y(x, y, s) = \iint_{\Omega} vf(u, v)dudv \quad (6)$$

$$I_{x^2+y^2}(x, y, s) = \iint_{\Omega} [u^2 + v^2] f(u, v)dudv. \quad (7)$$

The main point of using K as the filter kernel is that all the image moment integrals may be computed in constant time using the integral image technique: an integral image is precomputed for each image moment and is then used to obtain the required value. Therefore, the convolution F may be computed in $O(1)$ operations for any x, y and s .

To design a suitable filter whose response is expected to be close to the Gaussian filter, we need to choose proper values of A and B . This is done assuming that

1. K must be nonnegative within the support Ω , in order to have a smoothing filter,
2. $\iint_{\Omega} K(u, v)dudv = 1$,
3. There is a linear relation between the scale parameter and the standard deviation of the Gaussian filter, e.g. $s = C\sigma$.

To satisfy the first constraint we simply set $A = \frac{Bs^2}{2}$. The second one then gives directly the kernel normalization constant: $\iint_{\Omega} K(u, v)dudv = As^2 - \frac{Bs^4}{6} = \frac{Bs^4}{3}$. Dividing the kernel by this value and applying $s = C\sigma$ we obtain the filter kernel expression:

$$K(x, y) = \frac{3}{2s^2} - \frac{3}{s^4}(x^2 + y^2) = \frac{3}{C^2\sigma^2} \left(\frac{1}{2} - \frac{x^2 + y^2}{C^2\sigma^2} \right). \quad (8)$$

We finally tune the constant C to minimize the total squared difference between K and the Gaussian kernel, i.e. in order to assure the response close to the Gaussian one, obtaining $C \approx 3.5$. The resulting kernel K is shown in Figure 2.

3.2. Continuous response computation

To achieve subsample precision with the designed filter, a specific interpolation has to be applied to the integral images. The interpolation coefficients are derived assuming that the input image is a piecewise constant function having a constant value at each pixel position. Then,

- I_1 is piecewise constant,
- I_x and I_y are piecewise linear in x and y respectively,
- I_{x^2} and I_{y^2} are piecewise quadratic, $I_{x^2+y^2} = I_{x^2} + I_{y^2}$.

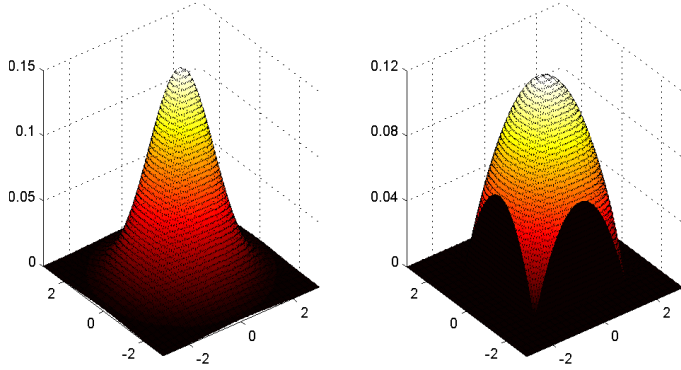


Fig. 2. Bi-dimensional Gaussian filter kernel of unit variance (left) compared to the proposed filter kernel (right). It provides a more accurate (closer to the Gaussian) output than the box filter, but the convolution may still be computed in $O(1)$ operations as the kernel surface is polynomial.

Image moment	ξ	η
I_1	α	β
I_x	$\alpha \frac{2x_0 + \alpha}{2x_0 + 1}$	β
I_y	α	$\beta \frac{2y_0 + \beta}{2y_0 + 1}$
I_{x^2}	$\alpha \frac{3\alpha x_0 + 3x_0^2 + \alpha^2}{3x_0^2 + 3x_0 + 1}$	β
I_{y^2}	α	$\beta \frac{3\beta y_0 + 3y_0^2 + \beta^2}{3y_0^2 + 3y_0 + 1}$

Table 1. Interpolation coefficients for integral images of the image moments.

It can be shown that the exact value of each image moment may be obtained through a linear interpolation with proper weights of the four closest neighbors of each vertex of the support Ω . Specifically, let $x = x_0 + \alpha$, $y = y_0 + \beta$, where $x_0 \in \mathbb{Z}$, $y_0 \in \mathbb{Z}$, and $0 \leq \alpha, \beta < 1$. Let J represent the integral image of a given image moment. Then

$$J(x, y) = [(1 - \xi)J(x_0, y_0) + \xi J(x_0 + 1, y_0)](1 - \eta) + [(1 - \xi)J(x_0, y_0 + 1) + \xi J(x_0 + 1, y_0 + 1)]\eta, \quad (9)$$

where the weights ξ and η in function of x , y , α and β are given in Table 1.

4. EXPERIMENTS AND DISCUSSION

In this section we compare the proposed approach notably to the original box filter. Our test data consists of several natural images acquired with a DSLR camera. In all our experiments, the box filter support size is defined as 2.6σ , which is a known convention when one tries to approximate the Gaussian. [14] As for the implementations of SIFT features and the separable convolution with Gaussian kernel, we take *VLFeat* library. [15]

4.1. Qualitative assessment

We first evaluate the proposed approach visually. Some filtered images are presented on Fig. 3. “Phantom contours” along the real edges in the box filter output appear due to the kernel discontinuity

near the support boundaries; their displacement is therefore related to the support size s . Typically, this kind of visual artifacts of low-pass filtering are undesirable. As the proposed filter has continuous falloff to the corners of Ω , the “phantom contours” are attenuated in the resulting image. However, they do not disappear completely as some discontinuities are present near middle points on the sides of the square support region (see Fig. 2).

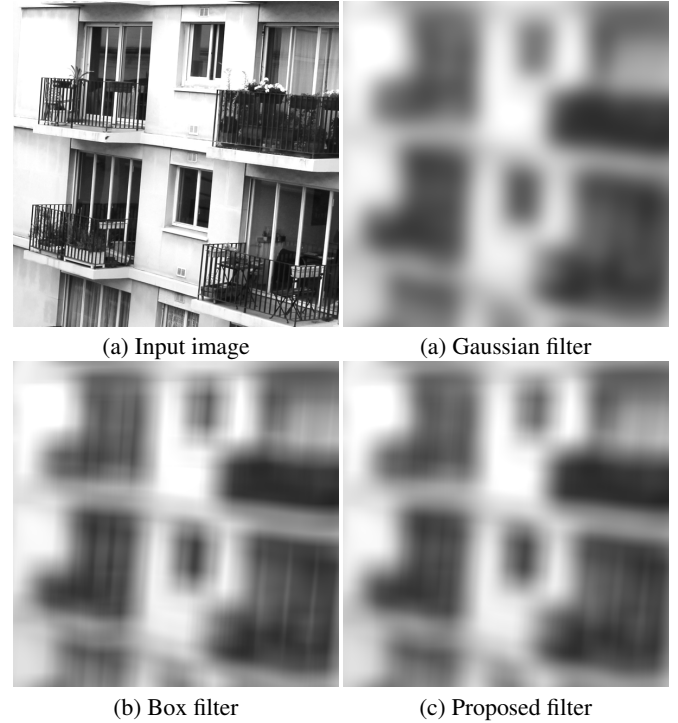


Fig. 3. An input image fragment of 600*600 pixels and filter outputs for $\sigma = 20$.

4.2. Rotational invariance

As it is discussed before, the invariance of the filter response to image rotations is a very desirable property in some cases. In this experiment we study the rotational invariance of our proposed filter. We proceeded as follows:

- an input image H is first smoothed to a level σ and stored to H_0 ,
- H is then rotated by angle a , smoothed with the same value of σ and then rotated back giving H_a ,
- H_0 is finally compared to H_a in a pixelwise manner. The difference between the images is evaluated in terms of PSNR. To avoid sampling artifacts at this point, σ is chosen large enough.

The experiment is repeated for several values of a from a given range.

A filter stable response to the image rotations will imply the rotated smoothed image H_a close to the smoothed image without rotations H_0 . The more invariant a filter is, the closer the two images should be. As it is explained before, the Gaussian filter reveals perfect rotational invariance.

We perform this experiment on a set of 5 different images (indoor and outdoor photos of 4.5 megapixels), smoothing each of them

Image resolution	σ	Gauss. ¹	Box		Proposed	
		t_S , ms	t_I , ms	t_S , ms	t_I , ms	t_S , ms
1.1 Mpix	1	86	13	34	63	124
	2	134				
	4	237				
	8	433				
	16	865				
4.5 Mpix	1	418	49	135	260	502
	2	611				
	4	994				
	8	1823				
	16	3543				
10.1 Mpix	1	968	110	300	579	1102
	2	1381				
	4	2264				
	8	4084				
	16	7905				

Table 2. Computation times for different filters in function of input image resolution and smoothing level σ . Each value is averaged over 10 repetitions.

with 3 values of σ : 5, 10 and 20, resulting in 15 cases per each value of the rotation angle α . The averaged results in function of α are presented on Fig. 4. The proposed filter clearly demonstrates a more stable response to image rotations than the box filter. Moreover, obtained PSNR values justify the visual similarity between the Gaussian and the proposed filter, which is illustrated on Fig. 3.

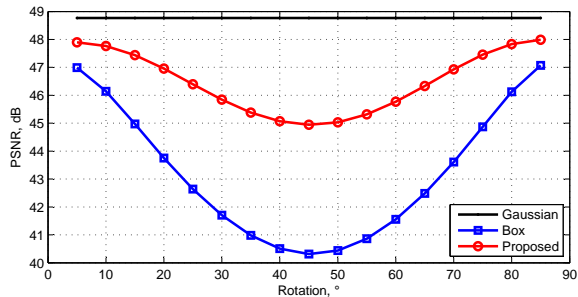


Fig. 4. Rotational invariance of the proposed filter vs the box filter. An image is rotated, smoothed and rotated back. The result is then compared to an image smoothed without rotations. Averaged results of 5 images and 3 levels of σ per each rotation angle are shown.

4.3. Computational time

In this section we evaluate the computational time of the proposed approach. The results for different values of σ and different input image resolutions are presented in Table 2. For each filter we present the smoothing time t_S and the initialization time t_I . The latter comprises the integral images computation.

4.4. Keypoint detection

We finally test the proposed filter in a keypoint detection scenario. Similarly to [7, 8] we replace the Gaussian scale space in the SIFT keypoint detector by image pyramids generated using our proposed filter and the box filter. Following a classic local feature evaluation

¹To have a fair comparison, we disabled the use of SSE2 instructions in *VLFeat* library in these tests.

procedure [16], we study the repeatability of detected local features obtained with different smoothing operators by means of *matching score*, i.e. ratio between the number of correctly matched features between two given images and the minimal number of detected features for these two images. We evaluate the features repeatability with respect to the in-plane image rotations. The resulting matching scores obtained with several different image sequences are presented on Fig. 5.

The proposed filter demonstrates stable repeatability gain with respect to the box filtering. The performance could be further improved by properly tuning the detector to the filter output.

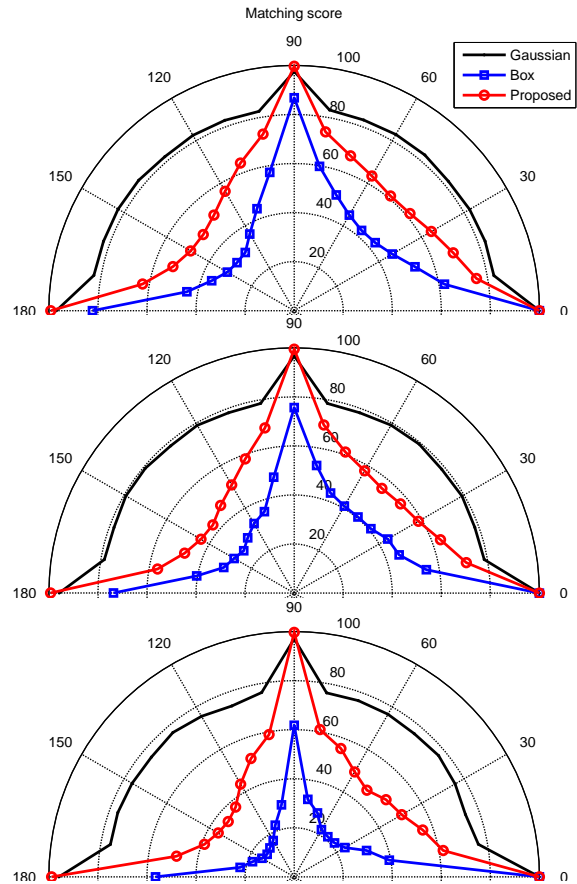


Fig. 5. Matching scores subject to in-plane rotations achieved by SIFT features detected with different filters. Three different image sequences of 1296*864 pixels representing in-plane rotations are matched against the corresponding upright images.

5. CONCLUSION

In this paper we proposed an efficient and accurate image smoothing operator, that provides a good trade-off between fast box filtering and classic Gaussian smoothing. Based on the integral images, our proposed filter inherits the ability of the box filter to compute the response in a constant time at any given point (x, y, σ) . This makes the approach particularly useful not only in feature matching applications, but also in cases where a non-structured smoothing is required, for example a non-uniform spatially adaptive filtering. Further efficiency improvements are possible, e.g., by using multi-threading for initialization or vectorial processor instructions.

6. REFERENCES

- [1] Firas Hassan, Nathan Pax, and Sami Khorbotly, "Reduced-latency architecture for image smoothing exponential filters," in *Proceed. IEEE Intern. Midwest Symposium on Circuits and Systems*, Fort Collins, Colorado, USA, August 2015.
- [2] David G Lowe, "Distinctive image features from scale-invariant keypoints," *Intern. J. of Comp. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [3] Joachim Weickert, *Anisotropic diffusion in image processing*, vol. 1, Teubner Stuttgart, 1998.
- [4] Tony Lindeberg and Lars Bretzner, "Real-time scale selection in hybrid multi-scale representations," *Scale Space Methods in Computer Vision*, pp. 148–163, 2003.
- [5] ISO/IEC JTC 1/SC 29/ WG 11, "CDVS: Requirements," MPEG document N11531, ISO/IEC, Geneva, July 2010.
- [6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (SURF)," *Comp. Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [7] Michael Grabner, Helmut Grabner, and Horst Bischof, "Fast approximated SIFT," in *Computer Vision–ACCV*. Hyderabad, India, January 2006.
- [8] Liang-Chi Chiu, Tian-Sheuan Chang, Jiun-Yen Chen, and Nelson Yen-Chung Chang, "Fast sift design for real-time visual feature extraction," *Proceed. of IEEE Intern. Conf. Image Proc.*, September 2013.
- [9] Konstantinos G Derpanis, Erich TH Leung, and Mikhail Sizintsev, "Fast scale-space feature representations by generalized integral images," in *Proceed. of IEEE Intern. Conf. Image Proc.*, San Antonio, TX, USA, September 2007.
- [10] Franklin C Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 207–212, 1984.
- [11] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [12] Ben Weiss, "Fast median and bilateral filtering," in *ACM SIGGRAPH 2006 Papers*, Boston, Massachusetts, 2006, SIGGRAPH '06.
- [13] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas, "Censure: Center surround extremas for realtime feature detection and matching," in *Computer Vision–ECCV*. Springer, Marseille, France, 2008.
- [14] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Barcelona, Spain, November 2011.
- [15] Andrea Vedaldi and Brian Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proceed. of Intern. Conf. on Multimedia*, New York, USA, 2010, MM '10.
- [16] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.