Accepted for publication on Pattern Recognition Letters, 2008

Improving the SIFT descriptor with smooth derivative filters

Plinio Moreno[a,*], Alexandre Bernardino[a], José Santos-Victor[a]

[a]*Instituto Superior Técnico & Instituto de Sistemas e Robótica*
Torre Norte Piso 7, Av. Rovisco Pais 1, 1049-001 Lisboa - Portugal

Abstract

Several approaches to object recognition make extensive use of local image information extracted in interest points, known as local image descriptors. State-of-the-art methods perform a statistical analysis of the gradient information around the interest point, which often relies on the computation of image derivatives with pixel differencing methods. In this paper we show the advantages of using smooth derivative filters instead of pixel differences in the performance of a well known local image descriptor. The method is based on the use of odd Gabor functions, whose parameters are selectively tuned to as a funcion of the local image properties under analysis. We perform an extensive experimental evaluation to show that our method increases the distinctiveness of local image descriptors for image region matching and object recognition.

Key words: Gabor filters, SIFT, local features, invariant descriptors, point matching, object recognition

1. Introduction

Successful image based object recognition methods recently developed are supported on the concept of local image descriptors. Image descriptors are localized information chunks extracted in particular points of the image, that remain stable in face of common image transformations, and with the ability to distinguish between different patterns. Several types of local descriptors have been reported in the literature[1, 2, 3, 4, 5, 6], but only recently a framework was proposed to compare their performance [7]. That work compares several types of image descriptors, such as differential operators, gradient histograms, correlation measures and image moments.

In the framework above mentioned, gradient based histograms showed the best performance. The initial steps consist in the interest point selection in scale space (e.g. Hessian, Harris), and the computation of the image gradients in the neighborhood of interest points (e.g. pixel differences, Canny detector). The descriptor is then obtained by splitting the interest point neighborhood into smaller regions (e.g. cartesian grid, log-polar grid), and finally for every subregion it is computed the histogram of the gradient orientation with an appropriate information selection procedure (e.g. weighting, Principal Component Analysis-PCA). To date, the most remarkable descriptor in terms of distinctiveness is the SIFT local descriptor [1], which computes the image gradient from pixel differences, subdivide the interest point regions in a cartesian grid, and for each subregion, compute the gradient orientation histogram weighted by the gradient magnitude. The descriptor is the concatenation of all subregion's histograms, followed by a unitary normalization.

Some extensions of the SIFT descriptor have been proposed recently, in order to improve either matching properties or reduce computational complexity. For instance, PCA-SIFT[2] concatenates the first order x and y image derivatives of every subregion, and for reducing the feature vector dimension is performed a PCA data selection. The main objective of PCA-SIFT is to keep the SIFT matching properties, reducing the descriptor size. On the other hand, the Gradient location-orientation histogram (GLOH) [7] is an extension of SIFT that computes the histogram using a log-polar spatial grid and reduces the descriptor size using PCA. The main objective of GLOH is to improve matching results by using a more robust spatial grid to compute the gradient histogram. Both PCA-SIFT and GLOH were tested in the comparison framework mentioned above [7] and have shown better performance than SIFT for some experimental conditions.

In this work we present an extension of the SIFT descriptor, proposing an alternative approach for gradient computation using smooth derivative filters. Using Gabor functions as smooth filters, our approach improves the distinctiveness of the SIFT local descriptor. In scale-normalized image regions, gradient computation using pixel differences, as in [1], is sensitive to noise and other artifacts induced by the image sensor and the normalization procedure. One common approach to diminish the noise sensitivity is to compute smoother approximations of the image derivatives using filters. In this work we use Gabor filters, which have been shown to approximate any image directional derivative [8], by suitable tuning their parameters. We propose a methodology to define the filters parameters based on local maxima of the magnitude of the filter response. We analyze the response for several filter widths, selecting the width in which the local maximum is located.

To evaluate the impact of our approach we use two evaluation frameworks: (i) Mikolajczyk local descriptor matching

*Corresponding author
Email addresses: plinio@isr.ist.utl.pt (Plinio Moreno), alex@isr.ist.utl.pt (Alexandre Bernardino), jasv@isr.ist.utl.pt (José Santos-Victor)

experiment [7], and (ii) an object recognition experiment. The first experiment evaluates the matching performance gain of our descriptor over SIFT descriptor, while the second experiment evaluates its impact in an object recognition task.

In the local descriptor evaluation framework, several types of images and image transformations are employed in the evaluation process. The procedure comprises five main steps: (i) Apply a specific transformation to each image in the data set and create pairs of images (original and transformed); (ii) in each pair of images find regions with suitable interest point detectors (e.g. blobs, corners, ridges) and corresponding normalization parameters (e.g. scale, rotation, affine); (iii) normalize regions to a fixed size and compute a set of local descriptors in the regions; (iv) for every pair of images, match the descriptors computed in the vicinity of the regions provided by the interest point detection procedure; (v) the evaluation criterion is composed by precision-recall curves of regions matched between two images. We utilize this evaluation framework to compare the distinctiveness of our descriptor proposal against SIFT descriptor.

In the object recognition experiment, we model object categories by a bunch of local descriptors, using an appearance only model, that disregards pose between local descriptors. We detect nine different object categories, considering each category as a two-class problem (object samples and background samples). Objects are modeled by a feature vector containing the similarity of the descriptors to one of the classes. In order to estimate class models, we use two learning algorithms: AdaBoost and SVM. In order to evaluate recognition performance, we compute the equal error point of the Receiver Operator Characteristic (ROC) curve for several object models. To build different object models, we vary: (i) the local descriptor, and (ii) the number of local descriptors that represent each category.

In Section 2 we explain the modification proposed to the SIFT descriptor. In Section 3 we describe in more detail the local descriptor evaluation framework. In Section 4 we describe the object recognition experiment in detail. The experimental results and discussion are included in Section 5, followed by conclusions in Section 6.

2. A local descriptor using smooth derivative filters

In this section we first review the SIFT local descriptor computation. Then we present a modification of the SIFT descriptor, using odd Gabor filters to compute first order image derivatives.

2.1. SIFT local descriptor

In the original formulation of the SIFT descriptor[1], a scale-normalized image region is represented with the concatenation of gradient orientation histograms relative to several rectangular subregions. First, to obtain the scale-normalized patches, a salient region detection procedure provides image point neighborhoods. The saliency function is the scale-space Difference of Gaussians (DoG), and the image regions (position and scale) are selected by the local extrema at DoG. In order to compute the local descriptor, the regions are scale normalized to compute the derivatives I_x and I_y of the image I with pixel differences:

$$I_x(x,y) = I(x+1,y) - I(x-1,y) \quad (1)$$

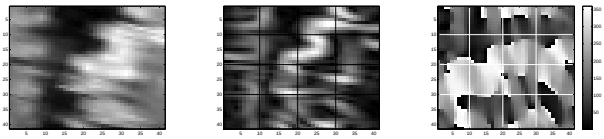
$$I_y(x,y) = I(x,y+1) - I(x,y-1). \quad (2)$$

Image gradient magnitude and orientation is computed for every pixel in the image region:

$$M(x,y) = \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \quad (3)$$

$$\Theta(x,y) = \tan^{-1}(I_y(x,y)/I_x(x,y)). \quad (4)$$

The interest region is then divided in subregions in a rectangular grid. In Figure 1 we see examples of the gradient magnitude and orientation for an image region and its corresponding 16 subregions (4 per dimension).



(a) normalized image region (b) $M(x,y)$ (c) $\Theta(x,y)$

Figure 1: Example of gradient magnitude and orientation images

The next step is to compute the histogram of gradient orientation, weighted by gradient magnitude, for each subregion. Orientation is divided into 8 bins and each bin is set with the sum of the windowed orientation difference to the bin center, weighted by the gradient magnitude:

$$h_{r_{(l,m)}}(k) = \sum_{x,y \in r_{(l,m)}} M(x,y)(1 - |\Theta(x,y) - c_k|/\Delta_k), \quad \Theta(x,y) \in \text{bin } k, \quad (5)$$

where c_k is the orientation bin center, Δ_k is the orientation bin width, and (x,y) are pixel coordinates in subregion $r_{(l,m)}$.

The SIFT local descriptor is the concatenation of the several gradient orientation histograms for all subregions:

$$u = (h_{r_{(1,1)}}, \ldots, h_{r_{(l,m)}}, \ldots, h_{r_{(4,4)}}) \quad (6)$$

The final step is to normalize the descriptor in Eq.(6) to unit norm, in order to reduce the effects of uniform illumination changes. The gradient orientation is not invariant to rotations of the image region, so the descriptor is not invariant. To provide orientation invariance, Lowe proposed to compute the orientation of the image region, and set the gradient orientation relative to the region's orientation. The orientation is given by the highest peak of the gradient orientation histogram of the image region. In further object recognition tests, we compute both invariant and non-invariant descriptors.

We have based our work in an approach similar to the one described here, proposing modifications only in the local descriptor computation. However, the gradient computation in the original SIFT descriptor is done with pixel differences which are very sensitive to noisy measurements. In next section we explain an alternative way to compute the image derivatives of Eq.(1) and Eq.(2), using smooth derivative filters where the degree of smoothing is appropriately selected.

2.2. Smooth derivative filters

The computation of image derivatives with pixel differences is an inherently noise sensitive process. Pixel differences implement a *high-pass* filtering operation on the image spectrum, amplifying the high frequency range, which is mainly composed by noise. To avoid such sensitivity, it is common to combine a *low-pass* filter (image blurring or smoothing) with the *high-pass* derivative filter, resulting in a *band-pass* filter, which we denote by *smooth derivative filter*. This effect can be implemented by either pre-smoothing the image followed by the derivative computation, or by convolving the image with a *band-pass* filter combining both phases. The important question to address at this point is "how much blurring should we apply to the image ?", or equivalently, "which frequency band should the *band-pass filter* focus on ?".

Several smooth derivative filters have been proposed for image filtering. Both Gaussian derivatives [8] and Gabor filters [9, 10] are common choices because of their properties and the availability of fast computation methods [11]. Gaussian derivatives [8] are smooth filters that can compute the image derivatives of any order. They have good noise attenuation properties due to an implicit image Gaussian filtering. On the other hand, Gabor filters are composed by Gaussian-modulated complex exponentials and provide an optimal trade-off between spatial and frequency resolution, allowing simultaneously good spatial localization and description of signal structures [9]. The application of Gabor functions to perform computer vision and image processing tasks has been motivated by biological findings in the low-level areas of primate visual cortex [12], and more recently by simulations of primate/human visual system [13, 14].

In this work we will use Gabor filters for the computation of smooth image derivatives due to the following facts:

- With appropriate parameters, odd Gabor filters can approximate odd order Gaussian directional derivatives [8].
- Gabor filters have a larger number of parameters than Gaussian derivatives, thus being more easily customized to each particular purpose. For instance, using the Gabor filter parameters in the edge cost function in order to find the appropiate filter parameters [15, 16].

Notice that the first fact listed above, tells us that the best performance with Gaussian derivative filters can also be achieved with Gabor filters, and the second fact suggests that a more careful parameter tuning of the Gabor parameters may eventually lead to better performance.

2.3. Gabor Filters for Image Derivative Computation

Gabor functions are defined by the multiplication of a complex exponential function (the carrier) and a Gaussian function (the envelope).

$$g_{x,y,\theta} = \frac{1}{2\pi\sigma_1\sigma_2} \cdot \cdot \exp\left(-\frac{(x\cos\theta + y\sin\theta)^2}{2\sigma_1^2} - \frac{(y\cos\theta - x\sin\theta)^2}{2\sigma_2^2}\right) \cdot \cdot \exp\left(i\frac{2\pi}{\lambda}(x\cos\theta + y\sin\theta)\right) \quad (7)$$

In the previous expression, (x, y) are the spatial coordinates, θ is the filter orientation, λ is its wavelength, and σ_1 and σ_2 are the Gaussian envelope standard deviations, oriented along directions θ and $\theta + \pi/2$, respectively.

To compute the first order image derivatives I_x and I_y we will use the odd (imaginary) part of the filter; the orientations will be $\theta = 0$ and $\theta = \pi/2$ for, respectively, the horizontal and vertical derivatives. To approximate the shape of an odd Gabor Filter to that of a Gaussian derivative, we set $\sigma_1 = \sigma_2 = \sigma$, and we introduce $\gamma = \lambda/\sigma$, a variable that is proportional to the number of wave periods within the filter width. By fixing an appropriate γ value, we will obtain an expression of the Gabor filter with a single parameter, the filter width σ.

If we look at the shape of the first order Gaussian derivatives at any scale in the derivative direction, there is one wave period within the spatial support of the filter, which roughly corresponds to $\lambda = 6\sigma$. Replacing this value in $\gamma = \frac{\lambda}{\sigma}$ yields $\gamma = 6$. By replacing $\sigma = \sigma_1 = \sigma_2$, and $\gamma = 6$ in Eq. (7), we obtain the filter being used in the remainder of the paper:

$$g_{x,y,\theta}(\sigma) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cdot \cdot \sin\left(\frac{2\pi}{6\sigma}(x\cos\theta + y\sin\theta)\right), \quad (8)$$

where $\theta = 0$ computes I_x, and $\theta = \pi/2$ computes I_y. The choice of σ will be done by an optimization procedure, based on the filter energy at locations with high gradient magnitude.

2.4. Scale-selection

In this section we propose a methodology to select a value for the scale parameter σ, such as to maximize the energy output of the smooth derivative filters in the analysis of the normalized regions obtained in the interest point selection procedure. We notice that, at this point, we have image regions that are already scale-normalized, therefore the scale-selection procedure we are proposing here should choose one single scale value for all regions.

In Figure 2 we see examples of the odd Gabor filter to compute the I_x at several σ values. In order to select the best scale we will use the gradient magnitude over all selected features in all images, due to its key role of weighting the gradient orientation histogram in the SIFT computation. In fact, the scale-normalized gradient magnitude has been used to measure edge

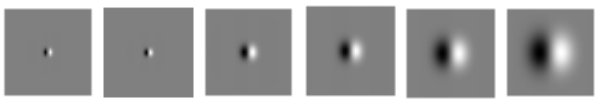


Figure 2: Examples of odd Gabor functions at $\theta = 0$, $\gamma = 6$, and $\sigma = \{2\sqrt{2}/3, 4/3, 4\sqrt{2}/3, 8/3, 8\sqrt{2}/3, 16/3\}$.

strength in scale-space [17]. However, this measure is not very stable for sufficiently large scales, leading to the selection of larger scale values in features with actual small scale [17]. This issue has been addressed in the context of edge scale selection, based on the concept of γ-normalized derivatives [18].

We have made some preliminary test with this methodology but the results were not promising, mainly because the features obtained in the interest point selection phase are not only edges, but also blobs, corners, junctions and other structures. Additionally, the image regions we are considering are already scale-normalized, so the scale selection procedure is a local search, as opposed to γ-normalized derivatives in [18]. We, therefore, propose the following methodology to avoid the bias toward large scales in the scale-normalized gradient magnitude:

- Considering independently the components of the normalized gradient magnitude. We have noticed that the horizontal and vertical derivatives are often better behaved than their combination in the gradient magnitude.
- Biasing the scale selection criterion to smaller scale values for each component, to avoid the non-decreasing behavior of the normalized derivatives for large scales [17].

Following these criteria, we pick the Gabor filter with largest energy in the x and y directions, and, from these, we select the smaller scale:

$$\hat{\sigma}_x = \arg\max_{\sigma} |(I * g_{x_i,y_i,\theta=0}(\sigma))| \quad (9)$$

$$\hat{\sigma}_y = \arg\max_{\sigma} |(I * g_{x_i,y_i,\theta=\pi/2}(\sigma))| \quad (10)$$

$$\hat{\sigma}(x_i, y_i) = \min(\hat{\sigma}_x, \hat{\sigma}_y), \quad (11)$$

$$I_x(x_i, y_i) = (I * g_{x_i,y_i,0}(\hat{\sigma}))(x_i, y_i) \quad (12)$$

$$I_y(x_i, y_i) = (I * g_{x_i,y_i,\pi/2}(\hat{\sigma}))(x_i, y_i). \quad (13)$$

where (x_i, y_i) is a point in the scale-normalized region, and $\hat{\sigma}$ is the adequate filter width at position (x_i, y_i).

2.4.1. Computational complexity

The local minima selection of Eqs. (9-13) has an obviously higher computational complexity than the pixel difference of Eqs. (1) and (2). In a scale-normalized image of size $S \times S$, the complexity of the pixel difference and filtering is $O(S^2)$, while the odd Gabor scale selection of Eqs. (9-13) have a complexity value of

$$O(S^2 \times (C \times F + 2F + 1)), \quad (14)$$

where C is the number of operations per pixel to compute the response of one Gabor filter and F is the number of Gabor filters applied. Using the state-of-the-art fast implementation of Gabor filters, $C = 60$[1] operations per pixel [19, 20], and F depends on the type of multi-scale filter implementation and the size of the normalized region. As we are dealing with scale-normalized regions, the search along F scales of Eqs. (9-13) can be replaced by a single scale suitable for all normalized images, thus yielding a complexity of $O(S^2 \times C)$.

In the following sections we describe how to evaluate the effect of the scale selection of smooth derivative filters of Eqs. (9-13) in SIFT performance. We compare performances using a very recent local descriptor evaluation framework.

3. Local descriptor evaluation

Recently was proposed a framework whose aim is to compare local image descriptors [7]. The method to compare is comprised by the steps we explain as follows:

Several image pairs are used for evaluation, each one having a particular type of image transformation. Each pair is obained by taking two pictures of the same scene in different conditions (position, camera/image settings). Figure 3 shows the test set images used to perform the local descriptor evaluation, the same as used in [7] for the sake of the comparison with the other methods. For each image, one of five possible image transformations is applied: Zoom + rotation, viewpoint, image blur, JPEG compression, and illumination. For viewpoint transformations, scale + rotation and image blur, two classes of images are considered: (i) *natural* images containing a large amount of randomly oriented textures; and (ii) *structured images* containing many distinctive long edge boundaries. In the case of JPEG compression and illumination transformations, only images from the *structured* type are employed.

For the generation of ground truth data (computing the correct matches between the two images), each pair of images is related by a projective transformation H. The homography is computed in two steps: (i) a first approximation to the homography is computed using manually selected points, then the transformed image is warped with this homography, and (ii) a robust small baseline homography estimation algorithm is used to compute the residual homography between the reference image and the warped one.

Salient image regions are computed using invariant region detectors. This process outputs elliptic regions in the two images that are good candidates for posterior matching. Four detectors are tested:

- the Harris-affine detector[21] computes corners and junctions covariant[2] to affine transformations up to a rotation factor;
- the Hessian-affine[22] detector computes blobs and ridges covariant to affine transformations up to a rotation factor;
- the Harris-laplace[23] detector computes corners and junctions covariant to scale and rotation changes; and

[1] Considering an isotropic and non-zero mean Gabor filter implementation

[2] Corresponding regions in the two images are called covariant.

- the Hessian-laplace[1, 21] detector computes blobs and ridges covariant to scale and rotation changes.

These methods provide not only the localization of the salient regions but also geometrical information regarding the size of the image region. Then, the region's dominant orientation is obtained by selecting the peak of the gradient histogram. With this information, each image region can be associated to an ellipse ($R(\mu)$) representing its dominant shape. Knowing the ground truth projective transformation H between the images, a *correspondence test* is proposed to evaluate the quality of the invariant image detection process. Two image regions $R(\mu_a)$ and $R(\mu_b)$ are corresponding if the overlap error is less than threshold ϵ_0,

$$1 - \frac{R(\mu_a) \cap R(H^T \mu_b H)}{R(\mu_a) \cup R(H^T \mu_b H)} < \epsilon_0. \quad (15)$$

In the previous equation $R(\mu)$ is the elliptic region defined by $x^T \mu x = 1$, where μ has the ellipse parameters, and H is the homography between images.

Candidate image regions are normalized for affine and illumination transformations using, respectively, the elliptic regions parameters computed in the previous steps, and image region graylevel statistics.

To represent the detected regions in a way suitable for matching, an extended description of its photometric properties must be provided. Each candidate image region is represented by the SIFT descriptor and our proposal. A *matching test* determines if two candidate regions (one on each image of the pair) are similar. Three different matching methods are employed: (i) thresholded euclidean distance between the two descriptors, (ii) nearest-neighbor, and (iii) nearest-neighbor distance ratio. Based on the ground truth data, matches are classified as correct or false. In the case of threshold-based matching, two descriptors u_a and u_b are matched if the euclidean distance is below a threshold. In the case of nearest neighbor, a match exists if u_b is the nearest neighbor to u_a and the euclidean distance between descriptors is below a threshold. In the case of nearest neighbor distance ratio, we have the descriptor u_a, the nearest neighbor u_b and the second nearest neighbor u_c. The descriptors u_a and u_b are matched if $||u_a - u_b||/||u_a - u_c|| < t$. The threshold-based method may assign several matches to one descriptor, while the other two method assign at most one match to each descriptor.

An evaluation metric is defined, based on precision and recall. $recall$ versus $1 - precision$ curve are computed for each image pair. The recall of the regions detected in two images is defined as:

$$recall = \frac{\#correct\ matches}{\#correspondences}. \quad (16)$$

The ratio between false matches and the total number of matches is given by $1 - precision$ value:

$$1 - precision = \frac{\#false\ matches}{\#correct\ matches + \#false\ matches}. \quad (17)$$

After completing the steps above, one is able to compare the matching performance of any local descriptor using the $recall$ versus $1 - precision$ curve. Additionally, we perform further tests in object category recognition, in order to compare the real performance of local descriptors in object recognition, and will be the subject of the next section.

4. Object Recognition Experiment

In this group of experiments we apply the appearance only model, in which objects are modeled by a bunch of local descriptors. The idea is to combine information of hundreds of thousands of local descriptors, being robust to occlusion and other noise sources [24, 25, 26]. This type of appearance only object model is adequate to compare the performance of local descriptors when matching objects, because it considers only local descriptor matches to find objects in new images.

The recognition of each object category is addressed as a two-class supervised problem (class label $c \in \{0, 1\}$), using positive (objects, $c = 1$) and negative (no objects, $c = 0$) class samples to learn the category model. The model consists of a class-similarity feature vector, that contains the matching to the descriptors of the class. The steps for the supervised learning of the model of an object category are as follows:

1. Select M interest points locations by applying the Difference of Gaussians (DoG) operator in the training set images $\{I_1^c, \ldots, I_t^c, \ldots, I_T^c\}$.
2. Compute a local descriptor at interest point locations, u_i^c, $i = 1, \ldots, M$, $c \in \{0, 1\}$.
3. Pick randomly $N << M$ interest points from the positive class samples as the category model. The respective picked local descriptors are denoted by

$$u_{s_1}, \ldots, u_{s_n}, \ldots, u_{s_N} \quad 1 \leq s_n \leq M \quad (18)$$

4. Compute the class-similarity feature vector $V_t^c = [v_1^c, \ldots, v_n^c, \ldots, v_N^c]$ for each image in the training set. Pick the descriptors u_i^c that belong to image I_t^c, and compute the similarity v_i^c of the descriptor u_{s_n}

$$v_n^c = \min_i \|u_{s_n} - u_i\|^2, \quad i = 1, \ldots, M, i \neq s_n, u_i^c \in I_t^c \quad (19)$$

5. Input the class similarity vectors $V_1^c, \ldots, V_t^c, \ldots, V_T^c$ with their respective label c to the learning algorithm, in order to estimate the category model.

After learning the object model, the steps to detect an instance of the object category in a new image are as follows:

1. Select J interest point locations.
2. Compute local descriptors in the new image $u_j, j = 1, \ldots, J$ at interest point locations.
3. Create class-similarity feature vector $V = [v_1, \ldots, v_n, \ldots, v_N]$ by matching each class model descriptor u_{s_n} against all descriptors u_j.

$$v_n = \min_i \|u_{s_n} - u_j\|^2, \quad j = 1, \ldots, J \quad (20)$$

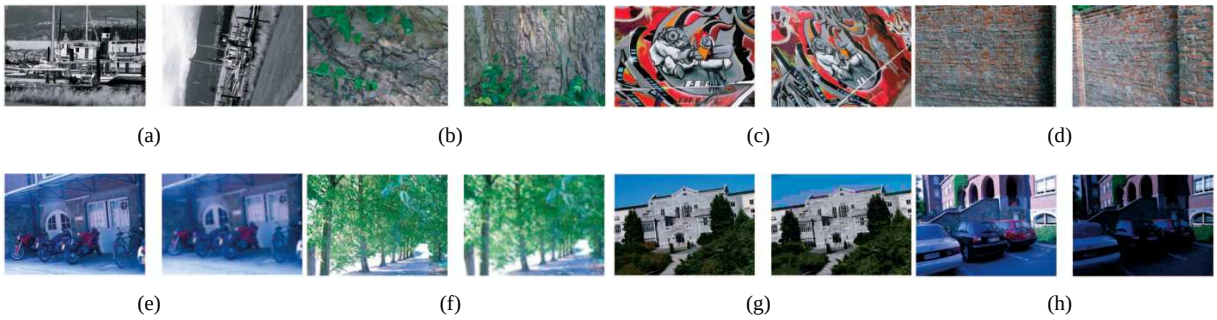


Figure 3: Data set used for local image descriptor evaluation. Zoom + rotation 3(a) and 3(b), viewpoint 3(c) and 3(d), image blur 3(e) and 3(f), JPEG compression 3(g), and illumination 3(h)

Figure 4: Typical images from selected databases.

4. Classify V as object or background image, with a binary classifier.

The experiments are performed over a set of classes provided by Caltech [3] : *airplanes side, cars side, cars rear, camels, faces, guitars, leaves, leopards* and *motorbikes side*, plus *Google things* dataset [27]. We use category *Google things* as negative samples. Each positive training set is comprised of 100 images drawn at random, and 100 images drawn at random from the unseen samples for testing. Figure 4 shows some sample images from each category. For all experiments, images have a fixed size (height 140 pixels), keeping the original image aspect ratio and converted to gray-scale format. We vary the number of local descriptors that represent an object category, $N = \{5, 10, 25, 50, 100, 250, 500\}$. In order to evaluate the influence of the learning algorithm, we utilize two classifiers: SVM [28] with linear kernel[4], and AdaBoost [30] with decision stumps. The evaluation criterion of every experiment is the performance at the equilibrium point of the Receiver Operator Characteristic (ROC) curve, when the false positive rate $=$ miss rate. Now we present the results when comparing the SIFT descriptor, and our smooth derivative SIFT, using the evaluation tools presented in this and the previous section.

5. Experimental results

We evaluate the impact of our proposed approach to compute the SIFT local descriptor in two related tasks: (i) image region matching, and (ii) component-based object recognition. First we present the experiments that allow the selection of a single scale value of the Gabor filter, in order to reduce the computational complexity of the image derivative method.

5.1. Gabor filter scale selection

Aiming to reduce the computational complexity presented in Eq. (14), we select a single filter suiting all cases. The single filter selection reduces the complexity of the image derivative computation from $O(S^2 \times (C \times F + 2F + 1))$ to $O(S^2 \times C)$. We compute the relative frequency (i.e. histogram) of the fiter width $\hat{\sigma}$ in Eq. (11), using all the scale-normalized image regions of the image data set presented in Figure 3. To avoid noisy $\hat{\sigma}$ values, we pick pixels with gradient magnitude above a certain threshold. We plot the marginalized (structured and textured) histograms and the total histogram in Figure 5. When comparing structured vs textured images, we observe that in the case of textured images the bins located at the left side of the histogram peak are all larger than the equivalent bins in the structured images histogram. This is an expected behavior, because the high gradient magnitude points in very textured images have a very small spatial support, while in structured images the points with high gradient magnitude have a larger spatial support. We also notice the difference of peak location between structured ($\hat{\sigma} = 1.88$) and textured ($\hat{\sigma} = 1.58$) images.

Although we biased the filter width selection to small values using Eq. (11), it still will select high filter width values in some of the image points (around 10% of image pixels), blurring the image gradient in some regions. This behaviour would lead to lose important histogram information in some subregions. In order to avoid these high filter width values, and keeping high frequency information of the textured images, we select the peak of the histogram of Figure 5 to compute the image derivatives. The image derivatives are given by

[3] Datasets are available at: http://www.robots.ox.ac.uk/~vgg/data3.html
[4] Implementation provided by *libsvm*[29]

Pixel difference of Eqs. (1-2)	0.44 ms
Multi-scale optimization (Gabor) of Eqs. (9-13)	9.75 ms
Single scale (Gabor) of Eqs. (21-22)	1.01 ms

Table 1: Execution time of C implementations, in a Pentium 4, 2.80 Ghz. Average value of the x derivative computation for all the normalized regions (size:41×41) selected in the images of Figure 3.

$$I_x(x,y) = (I * g_{x,y,0}(1.58))(x,y) \quad (21)$$
$$I_y(x,y) = (I * g_{x,y,\pi/2}(1.58))(x,y). \quad (22)$$

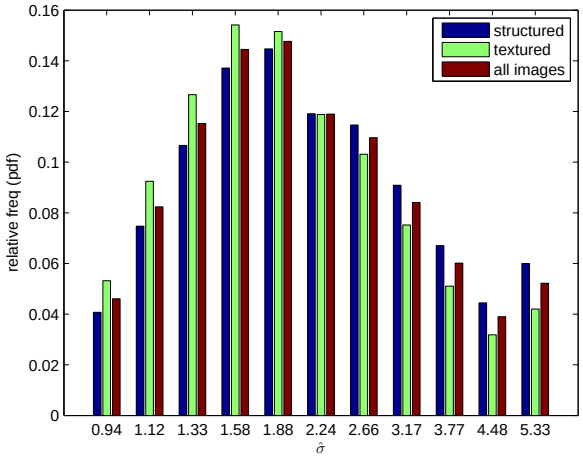


Figure 5: Histograms of $\hat{\sigma}$ for various image types.

The Eqs. (21, 22) provide a fast approximation of the scale selection of Eqs. (9-13), keeping the advantage of a smoother image derivative approximation versus the pixel differences of Eqs. (1) and (2). In the next sections we present the performance improvement of the SIFT descriptor by using Eqs. (21, 22). However, we pay the price of performance improvement by increasing the computational load of the image derivative computation, as shown in Table 1. Despite that the theoretical complexity analysis indicates a 60 times slow down with our approach, in practice we verified that it only slows down 2-3 times, thus maintaining a real-time functionality. The explanation may be related to the pixel access times to perform the subtraction, that were not considered in the theoretical analysis. Additionally, the fixed computational cost of the image normalization will further smooth out the differences between the two methods.

5.2. Image region matching

We compute $recall$ vs $1 - precision$ curves for all types of: (i) image transformations, (ii) image detectors, and (iii) structured and textured images. We observe in Figure 6 examples of the $recall$ vs $1 - precision$ curve in the case of the viewpoint transformation applied between the wall images of Figure 3(a), remarking that our descriptor curve is located above the original SIFT curve for the three matching criteria. We notice the same behavior for all the experiments, thus improving SIFT matching performance. In order to evaluate quantitatively the improvement of our descriptor over the original SIFT descriptor, in every experiment we compute the difference in recall rate for a fixed precision value of 0.5, obtaining the recall value by a linear interpolation using the two closest points.

	Harr	Hess	Struc	Text	Total
Threshold	2.7	4.3	3.7	2.3	3
NN	0.36	0.75	0.59	0.56	0.54
NN ratio	0.23	1.33	0.5	1.02	0.68

Table 2: Mean recall difference (%) between our SIFT descriptor and original SIFT [1], at $precision = 0.5$

We see in Table 2 that the improvement value depends on: (i) detectors and (ii) threshold criterion. Performance improvement of Hessian detectors is greater than Harris detectors for every matching criteria. Also the improvement depends highly on the matching criterion, as recall improvement in the threshold based method is about 10 times than the improvement in the nearest neighbor methods. This difference is related to the difficulty of improving the performance of the nearest neighbor methods, because demand a high precision rate with very few correspondences. Nevertheless, our method for computing SIFT local descriptor improves SIFT distinctiveness for all the matching experiments. Now we present the improvement results in the object recognition tests.

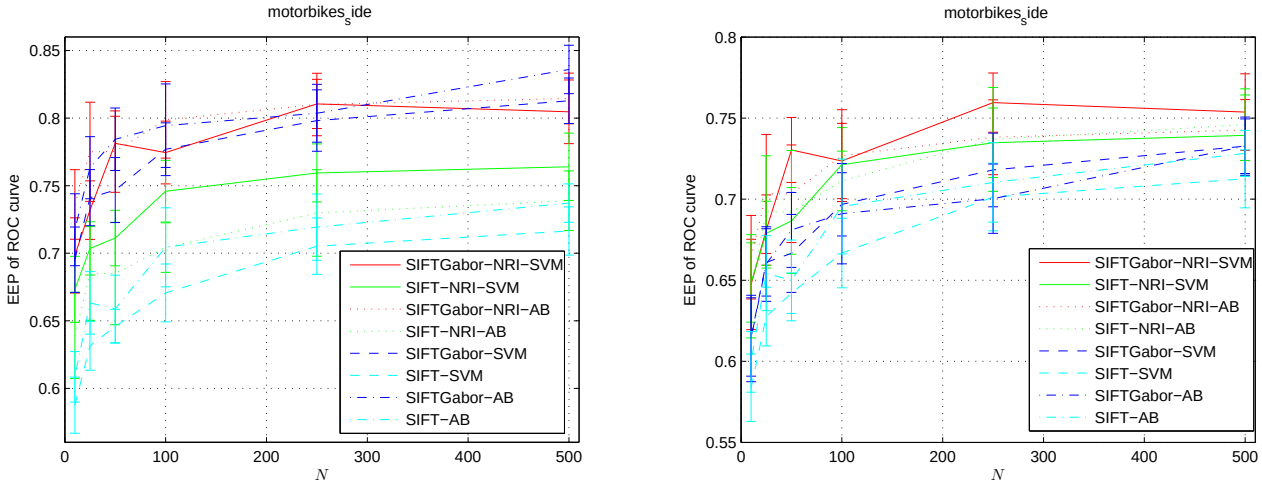
5.3. Component-based object detection

In this experiment we evaluate the impact of the performance improvement of our descriptor in an object category detection task. We test several variations of SIFT local descriptors to build the experimental set-up: (i) original SIFT (SIFT), (ii) original SIFT non-rotation-invariant (SIFT-NRI), (iii) modified SIFT (SIFTGabor), and (iv) modified SIFT non-rotation-invariant (SIFTGabor-NRI).

We compute the Equal Error Point (EEP) of the ROC curve of every type of: (i) Local descriptor, (ii) object category, and for (iii) two matching criteria: (a) threshold-based and (b) nearest neighbor. We see in Fig. 7 an example of the performance evolution as a function of the number of features, in the case of motorbikes category. This example shows the general result for all categories, where the best performance comprises: (i) SVM algorithm, (ii) NRI descriptors, (iii) threshold-based matching, and (iv) our modified SIFT descriptor.

	5	10	25	50	100	250	500
T	-0.58	0.36	1.51	2.76	1.22	0.89	0.46
NN	-0.66	-0.32	0.04	0.21	0.45	0.27	.07

Table 3: Mean difference (%) of the recognition rate at the EEP of ROC curve between our SIFT descriptor and original SIFT [1]. The middle and bottom row contain the results, respectively, for the threshold based (T) and the nearest neighbor (NN), for different number of descriptors N (shown in the top row).



(a) threshold-based matching

(b) nearest neighbor matching

Figure 7: Equal error point of ROC curve vs. number of local descriptors for the motorbike dataset, for various combinations descriptor-rotation option-classifier.

We observe in Table 3 the mean performance difference of: (i) all categories, (ii) non-rotation invariant descriptors, and (iii) SVM learning algorithm. The improvement of the threshold based criterion is at most 10 times the improvement of the nearest neighbor matching criteria.

Considering the matching criteria and the performance difference between our descriptor and original SIFT, we notice in general very similar results in this experiment to the image region matching results, having the best performance improvement in threshold-based criterion. We are able to maintain the performance improvement in a very challenging object recognition task, remarking the differences between both experiments in: (i) image datasets, and (ii) interest point detector (DoG vs. Hessian/Harris).

6. Conclusions

We present a modification of SIFT descriptor based on odd Gabor filters as smooth derivative filters, that improves the distinctiveness of SIFT descriptor. The modification proposed computes the first order image derivatives using odd Gabor filters as convolution kernels. The filters' parameters are selected by maximizing the filter response at locations with high image gradient. To evaluate the performance of our descriptor we perform two experiments: (i) image region matching, using Mikolajczyk and Schimd framework [7], and (ii) category object recognition, using a component-based model. In both experiments, our descriptor improves the SIFT distinctiveness.

The results of the image region matching experiments show that distinctiveness improvement is highly dependent on: (i) the matching criterion and (ii) the interest point detector. We obtain the best improvement results using the threshold-based matching criterion and Hessian-based interest point detectors.

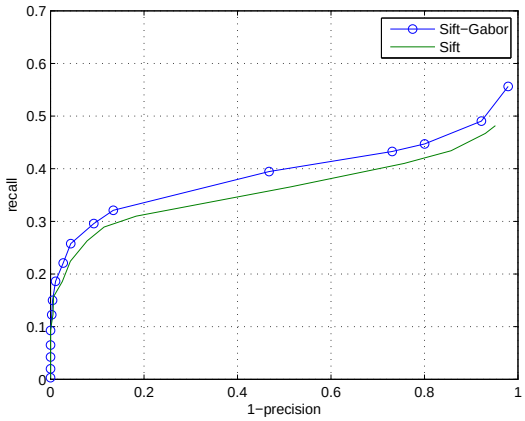
The object recognition experiment provides similar results, showing that recognition improvement depends on: (i) the learning algorithm used to classify objects, and (ii) the matching criterion used to build the feature vector. The best results are obtained using the SVM learning algorithm and the threshold-based matching criterion. The setup of the object recognition experiment presented in this work can be used to evaluate the impact of any kind of local descriptor, being able to compare local descriptor performances.

7. Acknowledgements

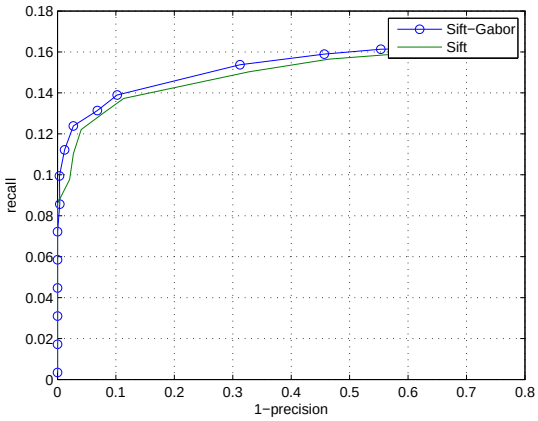
Research partly funded by the FCT Programa Operacional Sociedade de Informação(POSI) in the frame of QCA III, the Portuguese Foundation for Science and Technology Project 61911 - GESTINTERACT, the Portuguese Foundation for Science and Technology PhD Grant FCT SFRH\BD\10573\2002 and partially supported by Fundação para a Ciência e a Tecnologia (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds

References

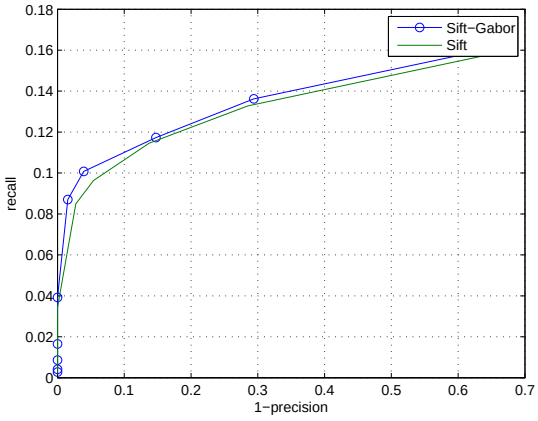
[1] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[2] Y. Ke, R. Sukthankar, Pca-sift: A more distinctive representation for local image descriptors, in: Proceedings IEEE CVPR, 2004, pp. 511–517.
[3] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE PAMI 13 (9) (1991) 891–906.
[4] J. Koenderink, A. van Doorn, Representation of local geometry in the visual system, Biological Cybernetics 55 (1987) 367–375.
[5] L. van Gool, T. Moons, D. Ungureanu, Affine/photometric invariants for planar intensity patterns, in: ECCV, 1996, pp. 642–651.
[6] P. Moreno, A. Bernardino, J. Santos-Victor, Gabor parameter selection for local feature detection, in: Proc. IbPRIA'05, Estoril, Portugal, 2005.
[7] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE PAMI 27 (10) (2005) 1615–1630.
[8] J. Koenderink, A. van Doorn, Generic neighborhood operators, IEEE PAMI 14 (6) (1992) 597–605.
[9] D. Gabor, Theory of communication, Journal of IEE 93 (1946) 429–459.



(a) threshold-based matching



(b) nearest neighbor matching



(c) nearest neighbor ratio matching

Figure 6: $recall$ vs. $1 - precision$ curves of Harris-affine regions matched using structured images in Figure 3(c), related by a viewpoint transformation.

[10] J. Daugman, Two-dimensional spectral analysis of cortical receptive fields profiles, Vision research 20 (1980) 847–856.

[11] I. Young, L. van Vliet, M. van Ginkel, Recursive gabor filtering, IEEE Transactions on Signal Processing 50 (11) (2002) 2798–2805.

[12] P. Daniel, D. Whitterridge, The representation of the visual field on the cerebral cortex in monkeys, Journal of Physiology 159 (1961) 203–221.

[13] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE PAMI 20 (11) (1998) 1254–1259.

[14] D. D. Deco, T. Lee, The role of early visual cortex in visual integration: a neural model of recurrent interaction, European Journal of Neuroscience 20 (2004) 1089–1100.

[15] K. Namuduri, R. Mehrotra, N. Ranganathan, Edge detection models based on gabor filters, in: Proc. of the 11th IAPR International Conference on Pattern Recognition, 1992, Vol. 3, 1992, pp. 729–732.

[16] Z. Zhu, H. Lu, Y. Zhao, Multi-scale analysis of odd gabor transform for edge detection, in: Proc. of the First International Conference on Innovative Computing, Information and Control, Vol. 2, 2006, pp. 578–581.

[17] T. Lindeberg, On scale selection for differential operators, in: Proc. 8th. Conference on Image Analysis, 1993, pp. 857–866.

[18] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, International Journal of Computer Vision 30 (2) (1998) 117–154.

[19] A. Bernardino, J. Santos-Victor, A real-time gabor primal sketch for visual attention, in: Proc. IbPRIA'05, Estoril, Portugal, 2005.

[20] A. Bernardino, J. Santos-Victor, Fast iir isotropic 2-d complex gabor filters with boundary initialization, IEEE Transactions on Image Processing 15 (11) (2006) 3338–3348.

[21] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, International Journal of Computer Vision 1 (60) (2004) 63–86.

[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. V. Gool, A comparison of affine region detectors, International Journal of Computer Vision 65 (1-2) (2005) 43–72.

[23] K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest points, in: Proc. of ICCV, 2001, pp. 525–531.

[24] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Object recognition with cortex-like mechanisms, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 411–426.

[25] A. Opelt, A. Pinz, M.Fussenegger, P.Auer, Generic object recognition with boosting, IEEE Transactions on Pattern Recognition and Machine Intelligence 28 (3).

[26] G. Csurka, C. Bray, C. Dance, L. Fan, Visual categorization with bags of keypoints, in: ECCV Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.

[27] R. Fergus, P. Perona, A. Zisserman, A sparse object category model for efficient learning and exhaustive recognition, in: CVPR, 2005, pp. 380–387.

[28] E. Osuna, R. Freund, F. Girosi, Support Vector Machines: training and applications., Tech. Rep. AI-Memo 1602, MIT (March 1997).

[29] C. Chang, C. Lin, LIBSVM: a library for support vector machines (April 2005).

[30] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Tech. rep., Dept. of Statistics. Stanford University (1998).