Published in Image Processing On Line on 2012–7–1.
ISSN 2105–1232 © 2012 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
http://demo.ipol.im/demo/82/

PREPRINT April 4, 2013

The Anatomy of the SIFT Method

Ives Rey Otero, Mauricio Delbracio

Abstract

This article presents a detailed analysis and implementation of the *Scale Invariant Feature Transform* (SIFT) [1], a popular image matching algorithm. SIFT is a complex chain of transformations; each element of this chain and the respective invariance properties are herein presented and analyzed. One of the main drawbacks of the SIFT algorithm is probably the large number of parameters that need to be set. This work contributes to a detailed dissection of this algorithm where a careful analysis of each of its design parameters is discussed and its impact shown in an online demonstration.

1 General description

The scale invariant feature transform, SIFT [1], transforms an image into a large set of compact descriptors. Each descriptor is formally invariant to an image *translation, rotation and zoom out*. SIFT descriptors have also proved to be robust to a wide family of image transformations, such as affine changes of viewpoint, noise, blur, contrast changes, scene deformation, while remaining discriminative enough for matching purposes.

The algorithm, as generally conceived, consists of two successive operations: the detection of interesting points (i.e., keypoints) and the extraction of a descriptor at each of them. Since these descriptors are robust, they are usually used for matching images. Although the comparison stage is not strictly within the SIFT algorithm, it is included in this paper for completeness.

The algorithm principle. From a multiscale representation of the image (i.e., a stack of images with increasing blur), SIFT detects a series of keypoints mostly in the form of blob-like structures and accurately locates their center (x, y) and their characteristic scale σ. Then, it computes the dominant orientation θ over a region surrounding each one of these keypoint. The knowledge of (x, y, σ, θ) permits to compute a local descriptor of each keypoint's neighborhood. From a normalized patch around each keypoint, SIFT computes a keypoint descriptor which is invariant to any translation, rotation and scale. The descriptor encodes the spatial gradient distribution around a keypoint by a 128-dimensional vector. This compact feature vector is used to match rapidly and robustly the keypoints extracted from different images.

The algorithmic chain. In order to attain scale invariance, SIFT builds on the Gaussian scale-space: a multiscale image representation simulating the family of all possible zooms out through increasingly blurred versions of the input image (see [2] for a gentle introduction to the subject). In this popular multiscale framework, the Gaussian kernel acts as an approximation of the optical blur introduced by a camera (represented by its point spread function). Thus, the Gaussian scale-space

can be interpreted as a stack of images, each of them corresponding to a different zoom factor. The Gaussian scale-space representation is presented in Section 2.

In order to produce translation and scale invariant descriptors, structures must be unambiguously located, both in scale and position. This excludes image edges and corners since they are translation or scale invariant structures and therefore cannot be linked to a specific triplet (x, y, σ). However, image blobs or more complex local structures characterized by their position and size, are the most suitable structures for SIFT.

The detection and location of keypoints is done by extracting the 3D extrema of a differential operator applied to the scale-space. The differential operator used in the SIFT algorithm is the difference of Gaussians (DoG), presented in Section 3.1. The extraction of 3D continuous extrema consists of two steps: first, the DoG representation is scanned for 3D discrete extrema. This gives a first coarse location of the continuous extrema, which are then refined to subpixel precision using a local quadratic model. The extraction of 3D extrema is detailed in Section 3.2. Since there are many phenomena that can lead to the detection of unstable keypoints, SIFT incorporates a cascade of tests to discard the less reliable ones. Only those that are precisely located and sufficiently contrasted are retained. Section 3.3 discuses two different discarding steps: the rejection of 3D extrema with small DoG value and the rejection of keypoint candidates laying on edges.

SIFT invariance to rotation is obtained by assigning a keypoint reference orientation. This reference is computed from the gradient orientation over a keypoint neighborhood. This is detailed in Section 4.1. Finally the spatial distribution of the gradient inside an oriented patch is encoded to produce the SIFT keypoint descriptor. The design of the SIFT keypoint invariant descriptor is described in Section 4.2. This ends the algorithmic chain that defines the SIFT algorithm.

Additionally, Section 5 illustrates how SIFT descriptors are typically used to find local matches between a pair of images. The method presented here is the matching procedure described in the original paper by D. Lowe.

This complex chain of transformation is governed by a large number of parameters. Section 6 summarizes the parameters of the SIFT algorithm and provides an analysis of their respective influence.

Table 1 summarizes the consecutive steps of the SIFT algorithm while the details of the adopted notation are presented in Table 2.

2 The Gaussian scale-space

The Gaussian scale-space representation is a stack of increasingly blurred images. This blurring process simulates the loss of detail produced when a scene is photographed from farther and farther (i.e. when the zoom-out factor increases). The scale-space, therefore, provides SIFT with scale invariance as it can be interpreted as the simulation of a set of snapshots of a given scene taken at different distances. In what follows we detail the construction of the SIFT scale-space.

2.1 Gaussian blurring

Consider a *continuous* image $u(\mathbf{x})$ defined for every $\mathbf{x} = (x, y) \in \mathbb{R}^2$. In this case, the continuous Gaussian smoothing is defined as the convolution

$$G_\sigma u(\mathbf{x}) := \int G_\sigma(\mathbf{x}')u(\mathbf{x} - \mathbf{x}')d\mathbf{x}'$$

where $G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2}e^{-\frac{\mathbf{x}^2}{2\sigma^2}}$ is the Gaussian kernel parameterized by its standard deviation $\sigma \in \mathbb{R}^+$. The Gaussian smoothing operator satisfies a semi-group relation. More precisely, the convolution

Stage	Description
1.	Compute the Gaussian scale-space in: image out: scale-space
2.	Compute the Difference of Gaussians (DoG) in: scale-space out: DoG
3.	Find candidate keypoints (3D discrete extrema of DoG) in: DoG out: $\{(x_d, y_d, \sigma_d)\}$ list of discrete extrema (position and scale)
4.	Refine candidate keypoints location with sub-pixel precision in: DoG and $\{(x_d, y_d, \sigma_d)\}$ list of discrete extrema out: $\{(x, y, \sigma)\}$ list of interpolated extrema
5.	Filter unstable keypoints due to noise in: DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
6.	Filter unstable keypoints laying on edges in: DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
7.	Assign a reference orientation to each keypoint in: scale-space gradient and $\{(x, y, \sigma)\}$ list of keypoints out: $\{(x, y, \sigma, \theta)\}$ list of oriented keypoints
8.	Build the invariant keypoints descriptor in: scale-space gradient and $\{(x, y, \sigma, \theta)\}$ list of keypoints out: $\{(x, y, \sigma, \theta, \mathbf{f})\}$ list of described keypoints

Table 1: Summary of the SIFT algorithm.

u	Images, defined on the continuous domain $(x, y) = \mathbf{x} \in \mathbb{R}^2$
$\mathbf{u}$	Digital images, defined in a rectangular grid $(m, n) \in \{0, \dots, M-1\} \times \{0, \dots, N-1\}$
v	Gaussian scale-space, defined on continuous domain $(\sigma, \mathbf{x}) \in \mathbb{R}^+ \times \mathbb{R}^2$
$\mathbf{v}$	Digital Gaussian scale-space, defined on octaves $\mathbf{v} = (\mathbf{v}^o)$, $o = 1, \dots, n_{\text{oct}}$ Each octave o is defined on a discrete grid $(s, m, n) \in \{0, \dots, n_{\text{spo}}+2\} \times \{0, \dots, M_o-1\} \times \{0, \dots, N_o-1\}$
w	Difference of Gaussians (DoG), defined on continuous domain $(\sigma, \mathbf{x}) \in \mathbb{R}^+ \times \mathbb{R}^2$
$\mathbf{w}$	Digital difference of Gaussians (DoG) defined on octaves $\mathbf{w} = (\mathbf{w}^o)$, $o = 1, \dots, n_{\text{oct}}$ Each octave o is defined on a discrete grid $(s, m, n) \in \{0, \dots, n_{\text{spo}}+1\} \times \{0, \dots, M_o-1\} \times \{0, \dots, N_o-1\}$
G_ρ	Continuous Gaussian convolution of standard deviation ρ
$\mathbf{G}_\rho$	Digital Gaussian convolution of standard deviation ρ (see eq. (2.4))
$\mathbf{S}_2$	Subsampling operator by a factor 2, $(\mathbf{S}_2\mathbf{u})(m, n) = \mathbf{u}(2m, 2n)$
$\mathbf{I}_\delta$	Digital bilinear interpolator by a factor $1/\delta$ (see Algorithm 2).

Table 2: Summary of the notation used in the article.

of u with two successive Gaussian kernels of standard deviations σ_1 and σ_2 can be computed as a Gaussian convolution of standard deviation $\sqrt{\sigma_1^2 + \sigma_2^2}$,

$$G_{\sigma_2}(G_{\sigma_1} u)(\mathbf{x}) = G_{\sqrt{\sigma_1^2+\sigma_2^2}} u(\mathbf{x}). \tag{2.1}$$

We call Gaussian scale-space of u the three-dimensional (3D) function

$$v : (\sigma, \mathbf{x}) \mapsto G_\sigma u(\mathbf{x}). \tag{2.2}$$

If u is continuous and bounded, v is the solution of the heat diffusion equation

$$\frac{\partial v}{\partial \sigma} = \sigma \Delta v, \tag{2.3}$$

with initial condition $v(0, \mathbf{x}) = u(\mathbf{x})$. This property will be useful to compute a differential operator on the Gaussian scale-space.

In the case of digital images there is some ambiguity on how to define a discrete counterpart to the continuous Gaussian smoothing operator. The original SIFT work of Lowe implements the digital Gaussian smoothing through a discrete convolution with a sampled and truncated Gaussian kernel.

Digital Gaussian smoothing. Let $\mathbf{g}_\sigma$ be the one-dimensional digital kernel obtained by sampling a truncated Gaussian function of standard deviation σ

$$\mathbf{g}_\sigma(k) = K e^{-\frac{k^2}{2\sigma^2}}, \quad -\lfloor 4\sigma \rfloor \leq k \leq \lfloor 4\sigma \rfloor$$

where $\lfloor \cdot \rfloor$ denotes the floor function and K is set so that $\sum_{-\lfloor 4\sigma \rfloor \leq k \leq \lfloor 4\sigma \rfloor} \mathbf{g}_\sigma(k) = 1$. Let $\mathbf{G}_\sigma$ denote the digital Gaussian convolution of parameter σ and $\mathbf{u}$ be a digital image of size $M \times N$. Its digital Gaussian smoothing, denoted $\mathbf{G}_\sigma \mathbf{u}$, is computed via a separable two-dimensional (2D) discrete convolution:

$$\mathbf{G}_\sigma \mathbf{u}(k, l) := \sum_{k'=-\lfloor 4\sigma \rfloor}^{+\lfloor 4\sigma \rfloor} \mathbf{g}_\sigma(k') \sum_{l'=-\lfloor 4\sigma \rfloor}^{+\lfloor 4\sigma \rfloor} \mathbf{g}_\sigma(l')\, \bar{\mathbf{u}}(k - k', l - l'), \tag{2.4}$$

where $\bar{\mathbf{u}}$ denotes the extension of $\mathbf{u}$ to $\mathbb{Z}^2$ via symmetrization with respect to -0.5, namely, $\bar{\mathbf{u}}(k, l) = \mathbf{u}(s_M(k), s_N(l))$ with $s_M(k) = \min(k \bmod 2M, 2M - 1 - k \bmod 2M)$.

For the range of values of σ considered in the described algorithm, the digital Gaussian smoothing operator satisfies a semi-group relation [3]. Applying successively two digital Gaussian smoothings of parameters σ_1 and σ_2 is equivalent to applying one digital Gaussian smoothing of parameter $\sqrt{\sigma_1^2 + \sigma_2^2}$,

$$\mathbf{G}_{\sigma_2}(\mathbf{G}_{\sigma_1} \mathbf{u}) = \mathbf{G}_{\sqrt{\sigma_1^2+\sigma_2^2}} \mathbf{u}. \tag{2.5}$$

2.2 Digital Gaussian scale-space

As previously introduced, the scale-space $v : (\mathbf{x}, \sigma) \mapsto G_\sigma u(\mathbf{x})$ is a set of increasingly blurred images, where the scale-space position $(\mathbf{x}, \sigma)$ refers to the pixel $\mathbf{x}$ in the image generated with blur σ. In what follows, we detail how to compute the *digital scale-space*, a discrete counterpart of the continuous Gaussian scale-space.

We will call digital scale-space a set of digital images with different levels of blur and different sampling rates, all of them derived from an input image $\mathbf{u}_{\text{in}}$ with an assumed blur level σ_{in}. This set is split into subsets where images share a common sampling rate. Since in the original SIFT algorithm the sampling rate is iteratively decreased by a factor of two, these subsets are called *octaves*. Let n_{oct} be the total number of octaves in the digital scale-space, $o \in \{1, \ldots, n_{\text{oct}}\}$ be the

index of each octave, and δ_o its inter-pixel distance. We will adopt as a convention that the input image $\mathbf{u}_{\text{in}}$ inter-pixel distance is $\delta_{\text{in}} = 1$. Thus, an inter-pixel distance $\delta = 0.5$ corresponds to a $2\times$ upsampling of this image while a $2\times$ subsampling results in an inter-pixel distance $\delta = 2$. Let n_{spo} be the number of scales per octave (the standard value is $n_{\text{spo}} = 3$). Each octave o contains the images $\mathbf{v}_s^o$ for $s = 1, \ldots, n_{\text{spo}}$, each of them with a different level of blur σ_s^o. The level of blur in the digital scale-space is measured taking as length unit the inter-sample distance in the sampling grid of the input image $\mathbf{u}_{\text{in}}$ (i.e. $\delta_{\text{in}} = 1$). The adopted configuration is illustrated in Figure 1.

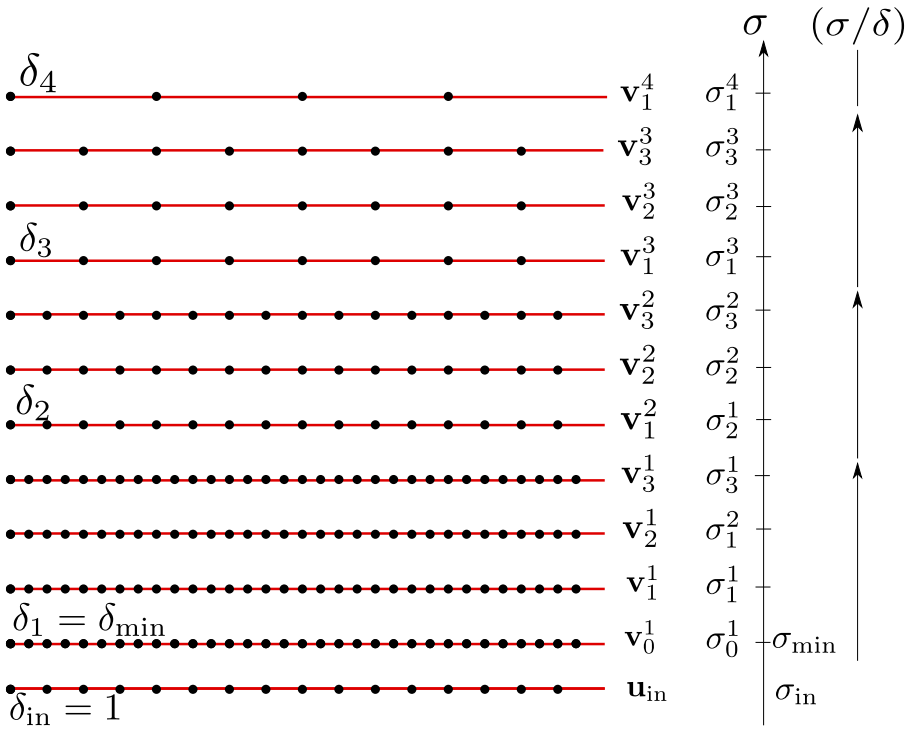


Figure 1: Convention adopted for the sampling grid of the digital scalespace $\mathbf{v}$. The level of blur is considered with respect to the sampling grid of the input image. The parameters are set to their standard value, namely $\sigma_{\text{min}} = 0.8, \sigma_{\text{min}} = 0.5, n_{\text{spo}} = 5, n_{\text{oct}} = 8, \sigma_{\text{in}} = 0.5$.

In practice, the digital scale-space will also include three additional images per octave, denoted by $\mathbf{v}_0^o, \mathbf{v}_{n_{\text{spo}}+1}^o, \mathbf{v}_{n_{\text{spo}}+2}^o$. The rationale for this will become clear later.

The construction of the digital scale-space begins with the computation of a *seed* image denoted $\mathbf{v}_0^1$. This image will have a blur level of $\sigma_0^1 = \sigma_{\min}$, which is the minimum level of blur considered, and a sampling rate $\delta_0 = \delta_{\min}$. It is computed from $\mathbf{u}_{\text{in}}$ by

$$\mathbf{v}_0^1 = \mathbf{G}_{\frac{1}{\delta_{\min}}\sqrt{\sigma_{\min}^2 - \sigma_{\text{in}}^2}} \mathbf{I}_{\delta_{\min}} \mathbf{u}_{\text{in}}, \tag{2.6}$$

where $\mathbf{I}_{\delta_{\min}}$ is the digital bilinear interpolator by a factor $1/\delta_{\min}$ (see Algorithm 1) and $\mathbf{G}_\sigma$ is the digital Gaussian convolution already defined. The entire digital scale-space is derived from this seed image. The standard value $\delta_{\min} = 0.5$ implies an initial $2\times$ interpolation. The blur level of the seed image, relative to the input image sampling grid, is usually set to $\sigma_{\min} = 0.8$.

The second and posterior scale-space images $s = 1, \ldots, n^{\text{spo}} + 2$ at each octave o are computed recursively according to

$$\mathbf{v}_s^o = \mathbf{G}_{\rho_{[(s-1)\to s]}} \mathbf{v}_{s-1}^o \tag{2.7}$$

where

$$\rho_{[(s-1)\to s]} = \frac{\sigma_{\min}}{\delta_{\min}} \sqrt{2^{2s/n^{\text{spo}}} - 2^{2(s-1)/n^{\text{spo}}}}.$$

The first images (i.e. $s = 0$) of the octaves $o = 2, \ldots, n^o$ are computed as

$$\mathbf{v}_0^o = \mathbf{S}_2 \mathbf{v}_{n^{\text{spo}}}^{o-1}, \tag{2.8}$$

where $\mathbf{S}_2$ denotes the subsampling operator by a factor of 2, $(\mathbf{S}_2\mathbf{u})(m, n) = \mathbf{u}(2m, 2n)$. This procedure produces a set of images $(\mathbf{v}_s^o)$, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n^{\text{spo}} + 2$, having inter-pixel distance

$$\delta_o = \delta_{\min} 2^{o-1} \tag{2.9}$$

and level of blur

$$\sigma_s^o = \frac{\delta_o}{\delta_{\min}} \sigma_{\min} 2^{s/n^{\text{spo}}}. \tag{2.10}$$

Consequently, the simulated blurs follow a geometric progression. The scale-space construction process is summarized in Algorithm 1. The digital scale-space architecture is thus defined by five parameters:

- the number of octaves n_{oct},
- the number of scales per octave n_{spo},
- the sampling distance $\delta_{\min}$ of the first image of the scale-space $\mathbf{v}_0^1$,
- the level of blur $\sigma_{\min}$ of the first image of the scale-space $\mathbf{v}_0^1$, and
- σ_{in} the assumed level of blur in the input image $\mathbf{u}^{\text{in}}$.

The diagram in Figure 2 depicts the digital scale-space architecture in terms of the sampling rates and levels of blur. Each point symbolizes a scale-space image $\mathbf{v}_s^o$ with inter-pixel distance δ^o and the level of blur σ_s^o. The featured configuration is produced from the original parameter values of the Lowe SIFT algorithm: $\sigma_{\min} = 0.8$, $\delta_{\min} = 0.5$, $n_{\text{spo}} = 3$, and $\sigma_{\text{in}} = 0.5$. The number of octaves n_{oct} is limited by the number of possible subsamplings. Figure 3 shows a subset of the digital scale-space images generated with the given scale-space configuration.

3 Keypoints detection

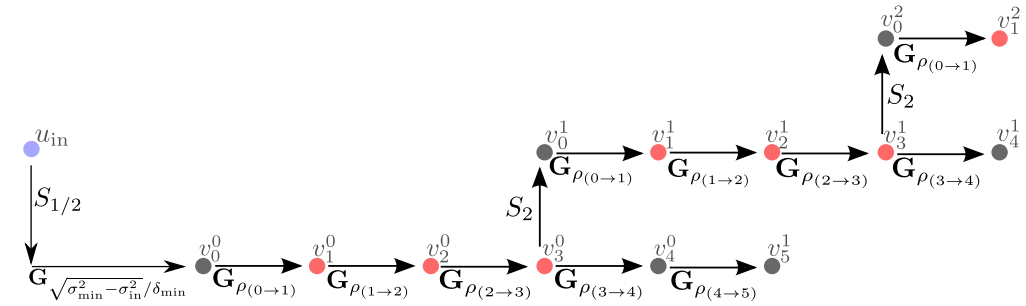
Differential operators are frequently used to extract features of interest from an image. Differential operators computed on a scale-space provide a keypoint location as well as its characteristic scale.

The extrema of the scale-space normalized Laplacian $\sigma^2\Delta v$ are the key features in the present framework. A Laplacian extremum is unequivocally characterized by its coordinates $(\sigma, \mathbf{x})$ in the scale-space where $\mathbf{x}$ refers to its center spatial position and σ relates to its size. As will be presented in Section 4, the knowledge of $(\sigma, \mathbf{x})$ enables the production of an invariant description of the extremum neighborhood. One possible solution for the detection of scale-space extrema is by computing the Laplacian of the image by a finite difference scheme. Instead, SIFT uses a difference of Gaussians operator (DoG) [4]. Let v be a scale-space and $\kappa > 1$. The difference of Gaussians (DoG) of ratio κ is defined by $w : (\sigma, \mathbf{x}) \mapsto v(\kappa\sigma, \mathbf{x}) - v(\sigma, \mathbf{x})$.

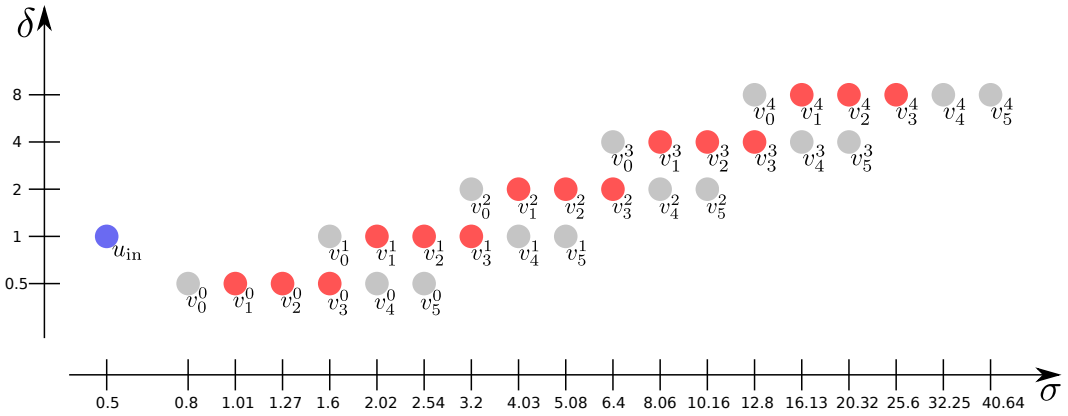
The DoG operator takes advantage of the link between the Gaussian kernel and the heat equation to efficiently compute an approximation of the normalized Laplacian. Indeed, from a set of simulated blurs following a geometric progression of ratio κ, the heat equation is approximated by

$$\sigma\Delta v = \frac{\partial v}{\partial \sigma} \approx \frac{v(\kappa\sigma, \mathbf{x}) - v(\sigma, \mathbf{x})}{\kappa\sigma - \sigma} = \frac{\omega(\sigma, \mathbf{x})}{(\kappa - 1)\sigma}.$$

Thus, we have $w(\sigma, \mathbf{x}) \approx (\kappa - 1)\sigma^2\Delta v(\sigma, \mathbf{x})$, the difference of Gaussians function ω approximates a constant factor of the normalized Laplacian $\sigma^2\Delta v$.



(a) Scalespace construction



(b) Scalespace standard configuration

Figure 2: (a) The succession of subsamplings and Gaussian convolutions producing the SIFT scale-space. The first image at each octave $\mathbf{v}_0^o$ is obtained via subsampling, with the exception of the first image at the first octave which is generated by a bilinear interpolation. (b) An illustration of the digital scale-space in its standard configuration. The digital scale-space $\mathbf{v}$ is composed of images $\mathbf{v}_s^o$ for $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 2$. All images are computed directly or indirectly from $\mathbf{u}^{\text{in}}$ (in blue). Each image is characterized by its level of blur and its sampling rate, respectively noted by σ and δ. The scale-space is split into octaves, namely sets of images sharing a common sampling rate. Each octave is composed of n_{spo} scales (in red) and other three auxiliary scales (in gray). The depicted configuration features $n_{\text{oct}} = 5$ octaves and corresponds to the following parameter settings: $n_{\text{spo}} = 3$, $\sigma_{\min} = 0.8$. The assumed level of blur of the input image is $\sigma_{\text{in}} = 0.5$.

Algorithm 1: Computation of the digital Gaussian scale-space

Input: $\mathbf{u}_{\text{in}}$, input digital image of $M \times N$ pixels.

Output: $(\mathbf{v}_s^o)$, digital scale-space, $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 2$.
$\mathbf{v}_s^o$ is a digital image of size $M_o \times N_o$, blur level σ_s^o (eq. (2.10)) and inter-pixel distance $\delta^o = \delta_{\min} 2^{o-1}$, with $M_o = \lfloor \frac{\delta_{\min}}{\delta_o} M \rfloor$ and $N_o = \lfloor \frac{\delta_{\min}}{\delta_o} N \rfloor$. The samples of $\mathbf{v}_s^o$ are denoted $\mathbf{v}_s^o(m, n)$.

Parameters: - n_{oct}, number of octaves.
- n_{spo}, number of scales per octave.
- $\sigma_{\min}$, blur level in the seed image.
- $\delta_{\min}$, inter-sample distance in the seed image.
- σ_{in}, assumed level of blur in the input image.

`//Compute the first octave`

`//Compute the seed image` $\mathbf{v}_0^1$

`//1.Interpolate original image (Bilinear interpolation, see Algo 2)`

$\mathbf{u}' \leftarrow$ `bilinear_interpolation`$(\mathbf{u}_{\text{in}}, \delta_{\min})$

`// 2. Blur the interpolated image (Gaussian blur, see eq (2.4))`

$\mathbf{v}_0^1 = \mathbf{G}_{\frac{1}{\delta_{\min}}\sqrt{\sigma_{\min}^2 - \sigma_{\text{in}}^2}} \mathbf{u}'$

`// Compute the other images in the first octave`

for $s = 1, \dots, n_{spo} + 2$ **do**
  $\mathbf{v}_s^1 = \mathbf{G}_{\rho_{[(s-1)\to s]}} \mathbf{v}_{s-1}^1$

`// Compute subsequent octaves`

for $o = 2, \dots, n_{oct}$ **do**
  `// Compute the first image in the octave by subsampling`
  for $m = 0, \dots, M_o - 1$ **and** $n = 0, \dots, N_o - 1$ **do**
    $\mathbf{v}_0^o(m, n) \leftarrow \mathbf{v}_{n^{\text{spo}}}^{o-1}(2m, 2n)$
  `// Compute the other images in octave` o
  for $s = 1, \dots, n_{spo} + 2$ **do**
    $\mathbf{v}_s^o = \mathbf{G}_{\rho_{[(s-1)\to s]}} \mathbf{v}_{s-1}^o$

Algorithm 2: Bilinear interpolation of an image

Input: $\mathbf{u}$, digital image, $M \times N$ pixels. The samples are denoted $\mathbf{u}(m, n)$.

Output: $\mathbf{u}'$, digital image, $M' \times N'$ pixels with $M' = \lfloor \frac{M}{\delta'} \rfloor$ and $N' = \lfloor \frac{N}{\delta'} \rfloor$.

Parameter: $\delta' < 1$, inter-pixel distance of the output image.

for $m' = 0, \dots, M' - 1$ **and** $n' = 0, \dots, N' - 1$ **do**
  $x = \delta' m'$
  $y = \delta' n'$

$$\begin{aligned}\mathbf{u}'(m', n') = &(x - \lfloor x \rfloor)\,((y - \lfloor y \rfloor)\,\bar{\mathbf{u}}(\lceil x \rceil, \lceil y \rceil) + (\lceil y \rceil - y)\,\bar{\mathbf{u}}(\lceil x \rceil, \lfloor y \rfloor)) \\ &+ (\lceil x \rceil - x)\,((y - \lfloor y \rfloor)\,\bar{\mathbf{u}}(\lfloor x \rfloor, \lceil y \rceil) + (\lceil y \rceil - y)\,\bar{\mathbf{u}}(\lfloor x \rfloor, \lfloor y \rfloor))\end{aligned}$$

where $\bar{\mathbf{u}}$ denotes the extension of $\mathbf{u}$ to $\mathbb{Z}^2$ via symmetrization withe respect to -0.5:
$\bar{\mathbf{u}}(k, l) = \mathbf{u}(s_M(k), s_N(l))$ with $s_N(k) = \min(k \bmod 2M, 2M - 1 - k \bmod 2M)$.

note: $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote respectively the floor and the ceil functions.

$\mathbf{v}_1^1$	$\mathbf{v}_2^2$	$\mathbf{v}_2^3$	$\mathbf{v}_2^4$	$\mathbf{v}_3^5$	$\mathbf{v}_5^5$
$\delta_1 = 0.5$	$\delta_2 = 1.0$	$\delta_3 = 2.0$	$\delta_4 = 4.0$	$\delta_5 = 8.0$	$\delta_5 = 8.0$
$\sigma_1^1 = 1.0$	$\sigma_2^2 = 2.5$	$\sigma_2^3 = 5.1$	$\sigma_2^4 = 10.2$	$\sigma_3^5 = 25.6$	$\sigma_5^5 = 40.6$

Figure 3: Crops of a subset of images extracted from the scale-space. The scale-space parameters are set to $n_{\text{spo}} = 3$, $\sigma_{\text{min}} = 0.8$, and the assumed input image blur level $\sigma_{\text{in}} = 0.5$. Image pixels are represented by a square of side δ_o for better visualization.

The SIFT keypoints of an image are defined as the 3D extrema of the difference of Gaussians (DoG). Since we deal with digital images, the continuous 3D extrema of the DoG cannot be directly computed. Nevertheless, we first detect the discrete extrema of the digital DoG and then refine their position. The detected points must be finally validated to discard possible unstable detections and false detections due to noise.

Hence, the detection of SIFT keypoints involves the following steps:

1. Compute the digital DoG.
2. Scan the digital DoG for 3D discrete extrema.
3. Refine position and scale of these candidates via a quadratic interpolation.
4. Discard unstable candidates such as uncontrasted candidates or candidates laying on edges.

We detail each of these steps in what follows.

3.1 Scale-space analysis: Difference of Gaussians

The digital DoG $\mathbf{w}$ is built from the digital scale-space $\mathbf{v}$. In each octave $o = 1, \dots, n_{\text{oct}}$ and for each image $\mathbf{w}_s^o$ with $s = 0, \dots, n_{\text{spo}} + 1$

$$\mathbf{w}_s^o(m, n) = \mathbf{v}_{s+1}^o(m, n) - \mathbf{v}_s^o(m, n)$$

with $m = 0, \dots, M_o - 1$, $n = 0, \dots, N_o - 1$. The image $\mathbf{w}_s^o$ will be linked to the level of blur σ_s^o. This computation is illustrated in Figure 4. See how, in the digital scale-space, the computation of the auxiliary scale $\mathbf{v}_{n_{\text{spo}}+2}^o$ is required for computing the DoG approximation $\mathbf{w}_{n_{\text{spo}}+1}^o$. Figure 5 illustrates the DoG scale-space $\mathbf{w}$ relative to the previously introduced scale-space $\mathbf{v}$.

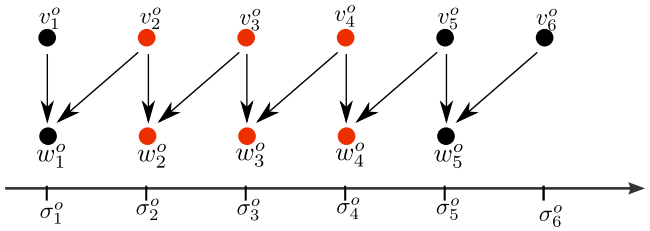


Figure 4: The difference of Gaussians operator is computed by subtracting pairs of contiguous images of the scale-space. The procedure is not centered: the difference between the images at scales $\kappa\sigma$ and σ is attributed a level of blur σ.

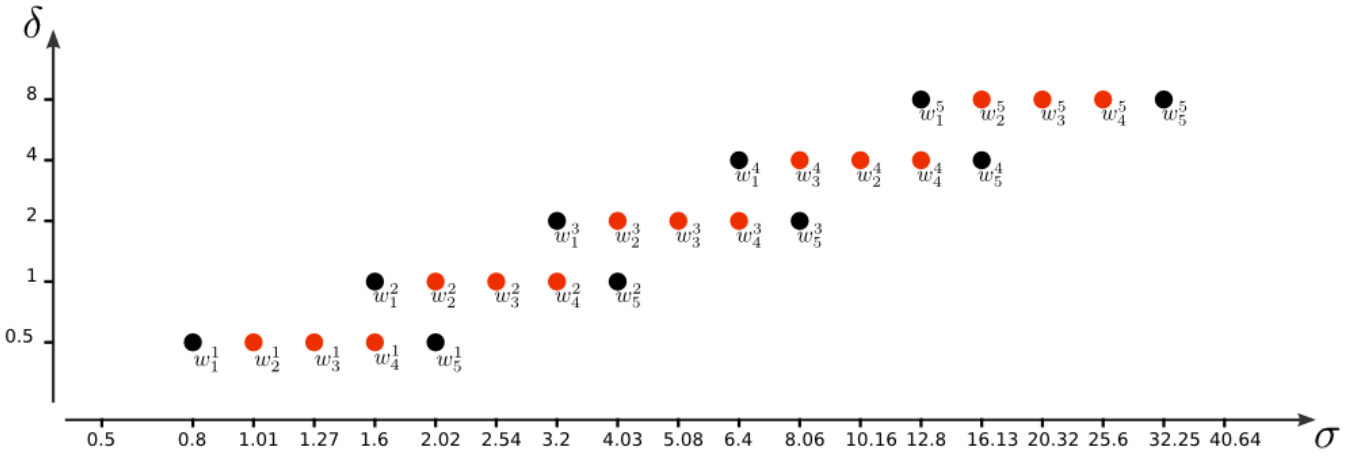


Figure 5: The *DoG* scale-space. The difference of Gaussians in an approximation of the normalized Laplacian $\sigma^2\Delta$. The difference $\mathbf{w}_s^o = \mathbf{v}_{s+1}^o - \mathbf{v}_s^o$ is relative to the level of blur σ_s^o. Each octave contains n_{spo} images plus two auxiliary images (in black).

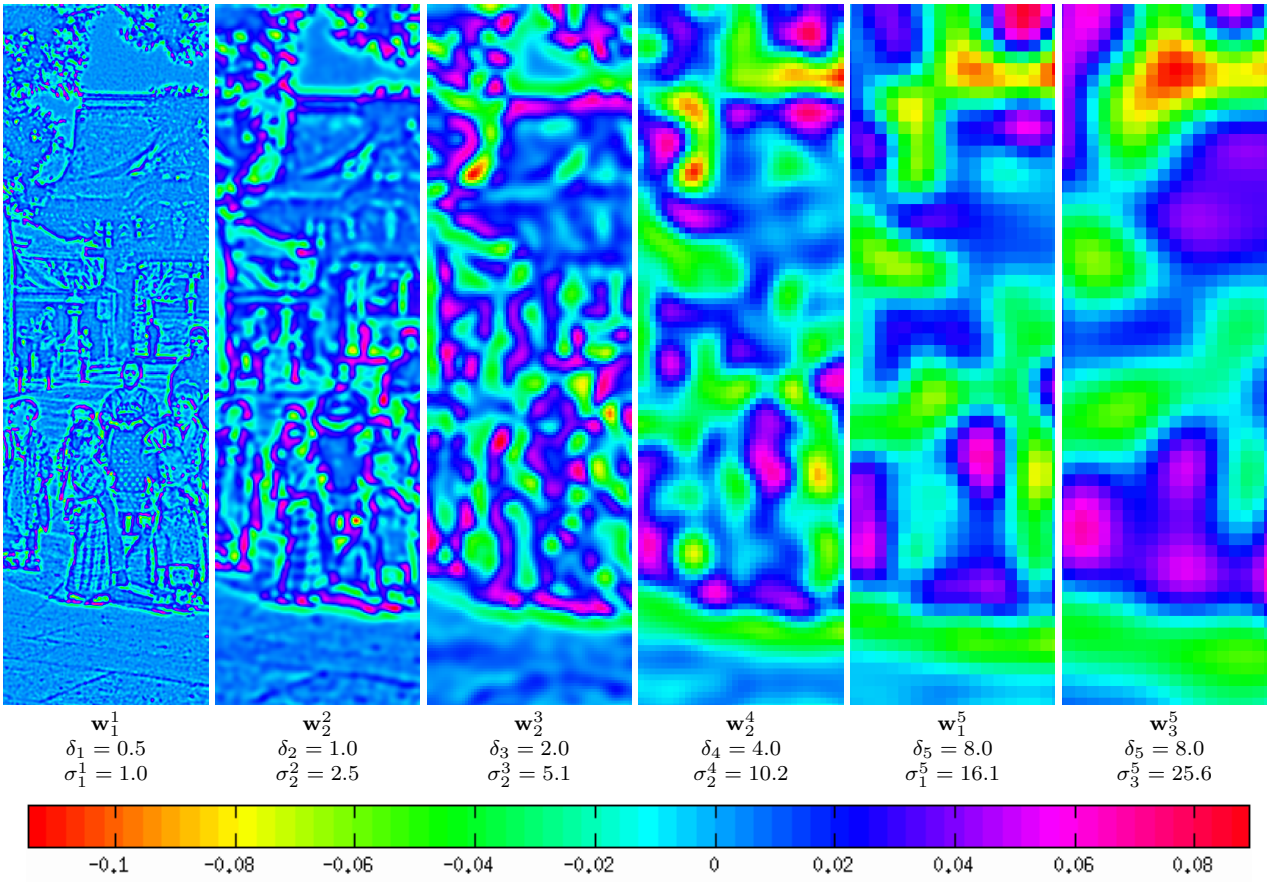


Figure 6: Crops of a subset of images extracted from the DoG space. The DoG operator is an approximation of the normalized Laplacian operator $\sigma^2 \Delta v$. The DoG scale-space parameters used in this example are as usual $n_{\text{spo}} = 3$, $\sigma_{\text{min}} = 0.8$, $\sigma_{\text{in}} = 0.5$. Image pixels are represented by a square of side δ_o for better visualization.

3.2 Extraction of candidate keypoints

Continuous 3D extrema of the digital DoG are calculated in two successive steps. The 3D discrete extrema are first extracted from $(\mathbf{w}_s^o)$ with pixel precision, then their location are refined through interpolation of the digital DoG by using a quadratic model. In the following, samples $\mathbf{v}_s^o(m,n)$ and $\mathbf{w}_s^o(m,n)$ are noted respectively $\mathbf{v}_{s,m,n}^o$ and $\mathbf{w}_{s,m,n}^o$ for better readability.

Detection of DoG 3D discrete extrema Each sample $\mathbf{w}_{s,m,n}^o$ of the DoG scale-space, with $s = 1,\dots,n_{\text{spo}}$, $o = 1,\dots,n_{\text{oct}}$, $m = 1,\dots,M_o - 2$, $n = 1,\dots,N_o - 2$ (which excludes the image borders and the auxiliary images) is compared to its neighbors to detect the 3D discrete maxima and minima (the number of neighbors is $26 = 3 \times 3 \times 3 - 1$). Note that these comparisons are possible thanks to the auxiliary images $\mathbf{w}_0^o$, $\mathbf{w}_{n_{\text{spo}}+1}^o$ calculated for each octave o. This scanning process is nevertheless a very rudimentary way to detect candidate points of interest. It is heavily subject to noise, produces unstable detections, and the information it provides regarding the location and scale may be flawed since it is constrained to the sampling grid. To amend these shortcomings, this preliminary step is followed by an interpolation that refines the localization of the extrema and by a cascade of filters that discard unreliable detections.

Keypoint position refinement At this stage, the location of each candidate keypoint is constrained to the sampling grid (defined by the octave o). Such coarse localization is an obstacle to reach full scale and translation invariance. SIFT refines the position and scale of each candidate keypoint using a local interpolation model.

Given a point (s,m,n) at the octave o in the digital DoG space, we denote by $\boldsymbol{\omega}_{s,m,n}^o(\boldsymbol{\alpha})$ the quadratic function at sample point (s,m,n) in the octave o, given by

$$\boldsymbol{\omega}_{s,m,n}^o(\boldsymbol{\alpha}) = \mathbf{w}_{s,m,n}^o + \boldsymbol{\alpha}^T \bar{g}_{s,m,n}^o + \frac{1}{2}\boldsymbol{\alpha}^T \bar{H}_{s,m,n}^o \boldsymbol{\alpha}, \tag{3.1}$$

where $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\alpha_3) \in [-0.5,0.5]^3$; $\bar{g}_{s,m,n}^o$ and $\bar{H}_{s,m,n}^o$ denote the 3D gradient and Hessian at (s,m,n) in the octave o, computed with finite difference schemes as follows:

$$\bar{g}_{s,m,n}^o = \begin{bmatrix} (\mathbf{w}_{s+1,m,n}^o - \mathbf{w}_{s-1,m,n}^o)/2 \\ (\mathbf{w}_{s,m+1,n}^o - \mathbf{w}_{s,m-1,n}^o)/2 \\ (\mathbf{w}_{s,m,n+1}^o - \mathbf{w}_{s,m,n-1}^o)/2 \end{bmatrix}, \quad \bar{H}_{s,m,n}^o = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{12} & h_{22} & h_{23} \\ h_{13} & h_{23} & h_{33} \end{bmatrix} \tag{3.2}$$

with

$$\begin{aligned} h_{11} &= \mathbf{w}_{s+1,m,n}^o + \mathbf{w}_{s-1,m,n}^o - 2.\mathbf{w}_{s,m,n}^o, & h_{12} &= (\mathbf{w}_{s+1,m+1,n}^o - \mathbf{w}_{s+1,m-1,n}^o - \mathbf{w}_{s-1,m+1,n}^o + \mathbf{w}_{s-1,m-1,n}^o)/4, \\ h_{22} &= \mathbf{w}_{s,m+1,n}^o + \mathbf{w}_{s,m-1,n}^o - 2.\mathbf{w}_{s,m,n}^o, & h_{13} &= (\mathbf{w}_{s+1,m,n+1}^o - \mathbf{w}_{s+1,m,n-1}^o - \mathbf{w}_{s-1,m,n+1}^o + \mathbf{w}_{s-1,m,n-1}^o)/4, \\ h_{33} &= \mathbf{w}_{s,m,n+1}^o + \mathbf{w}_{s,m,n-1}^o - 2.\mathbf{w}_{s,m,n}^o, & h_{23} &= (\mathbf{w}_{s,m+1,n+1}^o - \mathbf{w}_{s,m+1,n-1}^o - \mathbf{w}_{s,m-1,n+1}^o + \mathbf{w}_{s,m-1,n-1}^o)/4. \end{aligned}$$

This quadratic function can be interpreted as an approximation of the second order Taylor development of the underlying continuous function (where its derivatives are approximated by finite difference schemes).

In order to refine the position of a discrete extremum (s_e,m_e,n_e) at octave o_e we proceed as follows.

1. Initialize (s,m,n) by the discrete coordinates of the extremum (s_e,m_e,n_e).
2. Compute the continuous extrema of $\boldsymbol{\omega}_{s,m,n}^o$ by solving $\nabla\boldsymbol{\omega}_{s,m,n}^o(\boldsymbol{\alpha}) = 0$. This yields

$$\boldsymbol{\alpha}^* = -\left(\bar{H}_{s,m,n}^o\right)^{-1} \bar{g}_{s,m,n}^o. \tag{3.3}$$

3. If $\max(|\alpha_1^*|, |\alpha_2^*|, |\alpha_3^*|) < 0.5$ (i.e., the extremum of the quadratic function lies in its domain of validity) the extremum is accepted. According to the scale-space architecture (see Eq (2.10) and (2.9)), the corresponding keypoint coordinates are

$$(\sigma, x, y) = \left(\frac{\delta_{o_e}}{\delta_{\min}} \sigma_{\min} 2^{(\alpha_1^* + s)/n_{\text{spo}}} \,,\, \delta_{o_e}(\alpha_2^* + m) \,,\, \delta_{o_e}(\alpha_3^* + n) \right). \tag{3.4}$$

4. If $\boldsymbol{\alpha}^*$ falls outside the domain of validity, the interpolation is rejected and another one is carried out. Update (s, m, n) to the closest discrete value to $(s, m, n) + \boldsymbol{\alpha}^*$ and repeat from (2).

This process is repeated up to five times or until the interpolation is validated. If after five iterations the result is still not validated, the candidate keypoint is discarded. In practice, the validity domain is defined by $\max(|\alpha_1^*|, |\alpha_2^*|, |\alpha_3^*|) < 0.6$ to avoid possible numerical instabilities due to the fact that the piecewise interpolation model is not continuous. See Algorithm 6 for details.

According to the local interpolation model (3.1), the value of the DoG 3D interpolated extremum is

$$\begin{aligned} \boldsymbol{\omega} = \boldsymbol{\omega}^o_{s,m,n}(\boldsymbol{\alpha}^*) &= \mathbf{w}^o_{s,m,n} + (\boldsymbol{\alpha}^*)^T \bar{g}^o_{s,m,n} + \frac{1}{2}(\boldsymbol{\alpha}^*)^T \bar{H}^o_{s,m,n} \boldsymbol{\alpha}^* \\ &= \mathbf{w}^o_{s,m,n} - \frac{1}{2}(\bar{g}^o_{s,m,n})^T \left(\bar{H}^o_{s,m,n}\right)^{-1} \bar{g}^o_{s,m,n}. \end{aligned} \tag{3.5}$$

This value will be useful to assess the stability of the keypoint.

3.3 Filtering unstable keypoints

Discarding low contrasted extrema

Image noise will typically produce a large number of Laplacian extrema. Such extrema are normally unstable and are not linked to any particular structure in the image. SIFT attempts to eliminate these false detections by discarding candidate keypoints with a DoG value $\boldsymbol{\omega}$ below a threshold C_{DoG} (standard value $C_{\text{DoG}} = 0.03$ for $n_{\text{spo}} = 3$),

$$\textbf{if } |\boldsymbol{\omega}| < C_{\text{DoG}} \textbf{ then} \text{ discard the candidate keypoint.}$$

Since the DoG function approximates $(\kappa - 1)\sigma^2 \Delta v$, where κ is a function of the number of scales per octave n_{spo}, the value of threshold C_{DoG} will depend on the value of parameter n_{spo}. Before the refinement of the extrema, and in order to avoid unnecessary computations, a less conservative threshold at 80% of C_{DoG} is applied to the discrete 3D extrema,

$$\textbf{if } |\mathbf{w}^o_{s,m,n}| < 0.8 \times C_{\text{DoG}} \textbf{ then} \text{ discard the discrete 3D extremum.}$$

Discarding candidate keypoints on edges

In theory, perfect edges do not produce 3D DoG extrema. However, in practice, plenty of 3D discrete extrema are detected on edges. Some of these detections may even subsist after the interpolation refinement and the threshold on the DoG value. But as we have already pointed out, edges are not interesting structures for SIFT. Since they are translation invariant along the edge direction, they are poorly localized. Moreover, no reliable scale can be attributed to them. Hence, candidates keypoints laying on edges must be discarded.

The 2D Hessian of the DoG provides a characterization of those undesirable keypoint candidates. In terms of principal curvatures, edges present a large principal curvature orthogonal to the edge and a small one along the edge. In terms of the eigenvalues of the Hessian matrix, the presence of an edge amounts to a big ratio between the largest eigenvalue $\lambda_{\max}$ and the smallest one $\lambda_{\min}$.

The Hessian matrix of the DoG is computed at the nearest grid sample using a finite different scheme:

$$H^o_{s,m,n} = \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix}, \tag{3.6}$$

where

$$h_{11} = \mathbf{w}^o_{s,m+1,n} + \mathbf{w}^o_{s,m-1,n} - 2\mathbf{w}^o_{s,m,n}, \quad h_{22} = \mathbf{w}^o_{s,m,n+1} + \mathbf{w}^o_{s,m,n-1} - 2\mathbf{w}^o_{s,m,n},$$
$$h_{12} = h_{21} = (\mathbf{w}^o_{s,m+1,n+1} - \mathbf{w}^o_{s,m+1,n-1} - \mathbf{w}^o_{s,m-1,n+1} + \mathbf{w}^o_{s,m-1,n-1})/4.$$

The SIFT algorithm discards those keypoint candidates whose ratio of eigenvalues $r := \lambda_{\max}/\lambda_{\min}$ is less than a certain threshold C_{edge} (the standard value is $C_{\text{edge}} = 10$). Since only this ratio is relevant, the eigenvalues computation can be avoided by the following observation. The ratio of the Hessian matrix determinant and its trace are related to r by

$$\text{edgeness}(H^o_{s,m,n}) = \frac{\text{tr}(H^o_{s,m,n})^2}{\det(H^o_{s,m,n})} = \frac{(\lambda_{\max} + \lambda_{\min})^2}{\lambda_{\max}\lambda_{\min}} = \frac{(r+1)^2}{r}. \tag{3.7}$$

This is known as the *Harris-Stephen* edge response [5]. Thus, the filtering of keypoint candidates on edges consists in the following test:

$$\textbf{if } \text{edgeness}(H^o_{s,m,n}) > \frac{(C_{\text{edge}}+1)^2}{C_{\text{edge}}} \textbf{ then } \text{ discard candidate keypoint.}$$

Note that $H^o_{s,m,n}$ is the bottom-right 2×2 sub-matrix of $\bar{H}^o_{s,m,n}$ (3.2). Consequently the keypoint interpolation and the filtering of on-edge keypoints can be carried out simultaneously to save unnecessary computations.

3.4 Pseudocodes

Algorithm 3: Computation of the difference of Gaussians scale-space (DoG)

Input: $(\mathbf{v}^o_s)$, digital Gaussian scale-space, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n_{\text{spo}} + 2$.
Output: $(\mathbf{w}^o_s)$, digital DoG, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n_{\text{spo}} + 1$.

for $o = 1, \ldots, n_{oct}$ **and** $s = 0, \ldots, n_{spo} + 1$ **do**
  for $m = 0, \ldots, M_o - 1$ **and** $n = 0, \ldots, N_o - 1$ **do**
    $\mathbf{w}^o_s(m, n) = \mathbf{v}^o_{s+1}(m, n) - \mathbf{v}^o_s(m, n)$

Algorithm 4: Scanning for 3D discrete extrema of the DoG scale-space

Input: $(\mathbf{w}_s^o)$, digital DoG, $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 1$.
The samples of digital image $\mathbf{w}_s^o$ are denoted $\mathbf{w}_{s,m,n}^o$.

Output: $\mathcal{L}_{\text{A}} = \{(o, s, m, n)\}$, list of the DoG 3D discrete extrema.

for $o = 1, \dots, n_{oct}$ **do**
  for $s = 1, \dots, n_{spo}$, $m = 1, \dots, M_o - 2$ **and** $n = 1, \dots, N_o - 2$ **do**
    if *sample* $\mathbf{w}_{s,m,n}^o$ *is larger or smaller than all of its* $3^3 - 1 = 26$ *neighbors* **then**
      Add discrete extremum (o, s, m, n) to $\mathcal{L}_{\text{A}}$

Algorithm 5: Discarding low contrasted candidate keypoints (conservative test)

Inputs: - $(\mathbf{w}_s^o)$, digital DoG, $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 1$.
- $\mathcal{L}_{\text{A}} = \{(o, s, m, n)\}$, list of DoG 3D discrete extrema.

Output: $\mathcal{L}_{\text{A'}} = \{(o, s, m, n)\}$, filtered list of DoG 3D discrete extrema.
Parameter: C_{DoG} threshold.

for each *DoG* 3D *discrete extremum* (o, s, m, n) *in* $\mathcal{L}_A$ **do**
  if $|\mathbf{w}_{s,m,n}^o| \geq 0.8 \times C_{DoG}$ **then**
    Add discrete extremum (o, s, m, n) to $\mathcal{L}_{\text{A'}}$

Algorithm 6: Keypoints interpolation

Inputs: - $(\mathbf{w}_s^o)$, digital DoG scale-space, $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 1$.
- $\mathcal{L}_{\text{A'}} = \{(o, s, m, n)\}$, list of DoG 3D discrete extrema.

Output: $\mathcal{L}_{\text{B}} = \{(o, s, m, n, x, y, \sigma, \boldsymbol{\omega})\}$, list of candidate keypoints.

for each *DoG* 3D *discrete extremum* (o_e, s_e, m_e, n_e) *in* $\mathcal{L}_A$ **do**
  $(s, m, n) \leftarrow (s_e, m_e, n_e)$ `// initialize interpolation location`
  repeat
    `// Compute the extrema location and value of the local quadratic function` (see Algo 7)
    $(\boldsymbol{\alpha}^*, \boldsymbol{\omega}) \leftarrow$ `quadratic_interpolation`(o_e, s, m, n)
    `// Compute the corresponding absolute coordinates`
    $(\sigma, x, y) = \left(\frac{\delta_{o_e}}{\delta_{\min}} \sigma_{\min} 2^{(\alpha_1^* + s)/n_{\text{spo}}} \, , \, \delta_{o_e}(\alpha_2^* + m) \, , \, \delta_{o_e}(\alpha_3^* + n) \right)$.
    `// Update the interpolating position`
    $(s, m, n) \leftarrow ([s + \alpha_1^*], [m + \alpha_2^*], [n + \alpha_3^*])$
  until $\max(|\alpha_1^*|, |\alpha_2^*|, |\alpha_3^*|) < 0.6$ *or after* 5 *unsuccessful tries.*
  if $\max(|\alpha_1^*|, |\alpha_2^*|, |\alpha_3^*|) < 0.6$ **then**
    Add candidate keypoint $(o_e, s, m, n, \sigma, x, y, \boldsymbol{\omega})$ to $\mathcal{L}_{\text{B}}$

note: $[\cdot]$ denotes the round function.

Algorithm 7: Quadratic interpolation on a discrete DoG sample

Inputs: - $(\mathbf{w}_s^o)$, digital DoG scale-space, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n_{\text{spo}} + 1$.
- (o, s, m, n), coordinates of the DoG 3D discrete extremum.

Outputs: - $\boldsymbol{\alpha}^*$, offset from the center of the interpolated 3D extremum.
- $\boldsymbol{\omega}$, value of the interpolated 3D extremum.

Compute $\bar{g}^o_{s,m,n}$ and $\bar{H}^o_{s,m,n}$ `//DoG 3D gradient and Hessian by eq.`(3.2)

Compute $\boldsymbol{\alpha}^* = -\left(\bar{H}^o_{s,m,n}\right)^{-1} \bar{g}^o_{s,m,n}$

Compute $\boldsymbol{\omega} = \mathbf{w}^o_{s,m,n} - \frac{1}{2}(\bar{g}^o_{s,m,n})^T \left(\bar{H}^o_{s,m,n}\right)^{-1} \bar{g}^o_{s,m,n}$

Algorithm 8: Discarding low contrasted candidate keypoints

Input: $\mathcal{L}_{\text{B}} = \{(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})\}$, list of candidate keypoints.
Output: $\mathcal{L}_{\text{B'}} = \{(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})\}$, reduced list of candidate keypoints.
Parameter: C_{DoG} threshold.

for each *candidate keypoint* $(\sigma, x, y, \boldsymbol{\omega})$ *in* $\mathcal{L}_B$ **do**
  if $|\boldsymbol{\omega}| \geq C_{DoG}$ **then**
    Add candidate keypoint $(\sigma, x, y, \boldsymbol{\omega})$ to $\mathcal{L}_{\text{B'}}$.

Algorithm 9: Discarding candidate keypoints on edges

Inputs: - $(\mathbf{w}_s^o)$, DoG scale-space.
- $\mathcal{L}_{\text{B'}} = \{(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})\}$, list of candidate keypoints.
Output: $\mathcal{L}_{\text{C}} = \{(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})\}$, list of the SIFT keypoints.
Parameter: C_{edge}, threshold over the ratio between first and second Hessian eigenvalues.

for each *candidate keypoint* $(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})$ *in* $\mathcal{L}_{B'}$ **do**
  Compute $H^o_{s,m,n}$ by (3.6) `// 2D Hessian`
  Compute $\frac{\text{tr}(H^o_{s,m,n})^2}{\det(H^o_{s,m,n})}$ `// the Harris response`
  if $\frac{tr(H^o_{s,m,n})^2}{det(H^o_{s,m,n})} < \frac{(C_{edge}+1)^2}{C_{edge}}$ **then**
    Add candidate keypoint $(o, s, m, n, \sigma, x, y, \boldsymbol{\omega})$ to $\mathcal{L}_{\text{C}}$.

4 Keypoints description

In the literature, rotation invariant descriptors fall into one of two categories. On the one side those based on properties of the image that are already *rotation-invariant* and on the other side descriptors based on a normalization with respect to a reference orientation. The SIFT descriptor achieves rotation invariance by using the dominant gradient angle computed locally as a reference orientation, and then by normalizing the local gradient distribution with respect to this reference direction (see Figure 7).

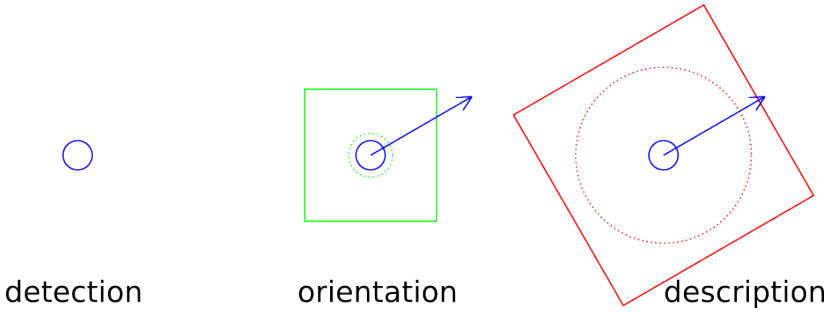


Figure 7: The description of a keypoint detected at scale σ (the radius of blue circle) consists of two local analysis of the gradient distribution covering different areas. **The first local analysis** aims at attributing a reference orientation to the keypoint (the blue arrow). It is performed over a Gaussian window of standard deviation $\lambda_{\text{in}}\sigma$ (the radius of the green circle). The patch $\mathcal{P}^{\text{ori}}$ (green square) of contributing samples has a width of $6\lambda_{\text{in}}\sigma$. The figure features the standard value for $\lambda_{\text{in}} = 1.5$. **The second analysis** aims at building the descriptor. It is performed over a Gaussian window of standard deviation $\lambda_{\text{descr}}\sigma$ (the radius of the red circle) within a square patch $\mathcal{P}^{\text{descr}}$ (the red square) of width of approximately $2\lambda_{\text{descr}}\sigma$. The figure features the standard settings : $\lambda_{\text{descr}} = 6$, with a Gaussian window of standard deviation 6σ and a patch $\mathcal{P}^{\text{descr}}$ of width 15σ.

The SIFT descriptor is built from the normalized image gradient orientation in the form of quantized histograms. In what follows, we describe how the reference orientation specific to each keypoint is defined and computed.

4.1 Keypoint reference orientation

The dominant gradient orientation over a keypoint neighborhood is used as its reference orientation. This allows for orientation normalization and hence rotation-invariant description (see Figure 7). Measuring this reference orientation involves three steps:

A. The local distribution of the gradient angle within a normalized patch is accumulated in an orientation histogram.

B. The orientation histogram is smoothed.

C. One or more reference orientations are extracted from the smoothed histogram.

A. Orientation histogram accumulation. Given an interpolated keypoint (x, y, σ), the patch to be analyzed is extracted from the image of the scale-space $\mathbf{v}_s^o$, whose σ_s^o is nearest to σ. This normalized patch, noted $\mathcal{P}^{\text{ori}}$, is the set of pixels (m, n) of $\mathbf{v}_s^o$ satisfying:

$$\max(|\delta_o m - x|, |\delta_o n - y|) \leq 3\lambda_{\text{ori}}\sigma. \tag{4.1}$$

The orientation histogram h from which the dominant orientation is found covers the range $[0, 2\pi]$. It is composed of n^{bins} bins with centers $\theta_k = 2\pi k/n^{\text{bins}}$. Each pixel (m, n) in $\mathcal{P}^{\text{ori}}$ will contribute to the histogram with a total weight of $c_{m,n}^{\text{ori}}$, which is the product of the gradient norm and a Gaussian

weight of standard deviation $\lambda_{\text{ori}}\sigma$ (standard value $\lambda_{\text{ori}} = 1.5$) reducing the contribution of distant pixels.

$$c^{\text{ori}}_{m,n} = \frac{1}{\sqrt{2\pi}\lambda_{\text{ori}}\sigma} e^{-\frac{\|(m\delta_o, n\delta_o)-(x,y)\|^2}{2(\lambda_{\text{ori}}\sigma)^2}} \left\| \left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) \right\|. \tag{4.2}$$

This contribution is assigned to the nearest bin, namely the bin of index

$$b^{\text{ori}}_{m,n} = \left[\frac{n^{\text{bins}}}{2\pi} \left(\text{arctan2}\left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) \bmod 2\pi \right) \right]. \tag{4.3}$$

where $[\cdot]$ denotes the round function. The gradient components of the scale-space image $\mathbf{v}^s_o$ are computed through a finite difference scheme

$$\partial_m \mathbf{v}^o_{s,m,n} = \frac{1}{2}\left(\mathbf{v}^o_{s,m+1,n} - \mathbf{v}^o_{s,m-1,n}\right), \qquad \partial_n \mathbf{v}^o_{s,m,n} = \frac{1}{2}\left(\mathbf{v}^o_{s,m,n+1} - \mathbf{v}^o_{s,m,n-1}\right), \tag{4.4}$$

for $m = 1, \ldots, M_o - 2$ and $n = 1, \ldots, N_o - 2$.

B. Smoothing the histogram. After being accumulated and before being analyzed, the orientation histogram is smoothed by applying six times a circular convolution with the three-tap box filter $[1, 1, 1]/3$.

C. Extraction of reference orientation(s). Keypoint reference orientations correspond to local histogram maxima larger than t times the histogram's maximum value with $t < 1$ (standard value $t = 0.8$). Let $k \in \{1, \ldots, n^{\text{bins}}\}$ be the index of a bin such that $h_k > h_{k^-}$, $h_k > h_{k^+}$ ($k^- = (k-1) \bmod n^{\text{bins}}$ and $k^+ = (k+1) \bmod n^{\text{bins}}$) and such that $h_k \geq t \max(h)$. This bin is centered on orientation $\theta_k = \frac{2\pi(k-1)}{n^{\text{bins}}}$. The corresponding keypoint reference orientation θ_{ref} is computed from the maximum position of the quadratic function that interpolates the values h_{k^-}, h_k, h_{k^+},

$$\theta_{\text{ref}} = \theta_k + \frac{\pi}{n^{\text{bins}}} \left(\frac{h_{k^-} - h_{k^+}}{h_{k^-} + 2h_k + h_{k^+}} \right). \tag{4.5}$$

Each one of the extracted reference orientations leads to the computation of one invariant local descriptor of a keypoint neighborhood. Note that consequently the number of descriptors may exceed the number of keypoints.

4.2 Keypoint normalized descriptor

The local descriptor of each keypoint neighborhood is designed to be invariant to translation, zoom and rotation. It describes the local spatial distribution of the gradient orientation over a normalized neighborhood. Given a detected keypoint, the normalized neighborhood consists in a square patch centered on the keypoint and aligned with the reference orientation. The descriptor consists in a set of orientation weighted histograms, each located on a portion of the square patch.

The normalized patch For each keypoint $(x_{\text{key}}, y_{\text{key}}, \sigma_{\text{key}}, \theta_{\text{key}})$, a normalized patch is isolated inside the image relative to the nearest discrete scale (o, s) to scale σ_{key}, namely $\mathbf{v}^o_s$. Any sample (m, n) in $\mathbf{v}^o_s$, of coordinates $(x_{m,n}, y_{m,n}) = (m\delta^o, n\delta^o)$ with respect to the sampling grid of the input image, has normalized coordinates $(\hat{x}_{m,n}, \hat{y}_{m,n})$ with respect to the keypoint $(x_{\text{key}}, y_{\text{key}}, \sigma_{\text{key}}, \theta_{\text{key}})$,

$$\begin{aligned} \hat{x}_{m,n} &= \left((m\delta_o - x_{\text{key}})\cos\theta_{\text{key}} + (n\delta_o - y_{\text{key}})\sin\theta_{\text{key}}\right)/\sigma_{\text{key}}, \\ \hat{y}_{m,n} &= \left(-(m\delta_o - x_{\text{key}})\sin\theta_{\text{key}} + (n\delta_o - y_{\text{key}})\cos\theta_{\text{key}}\right)/\sigma_{\text{key}}. \end{aligned} \tag{4.6}$$

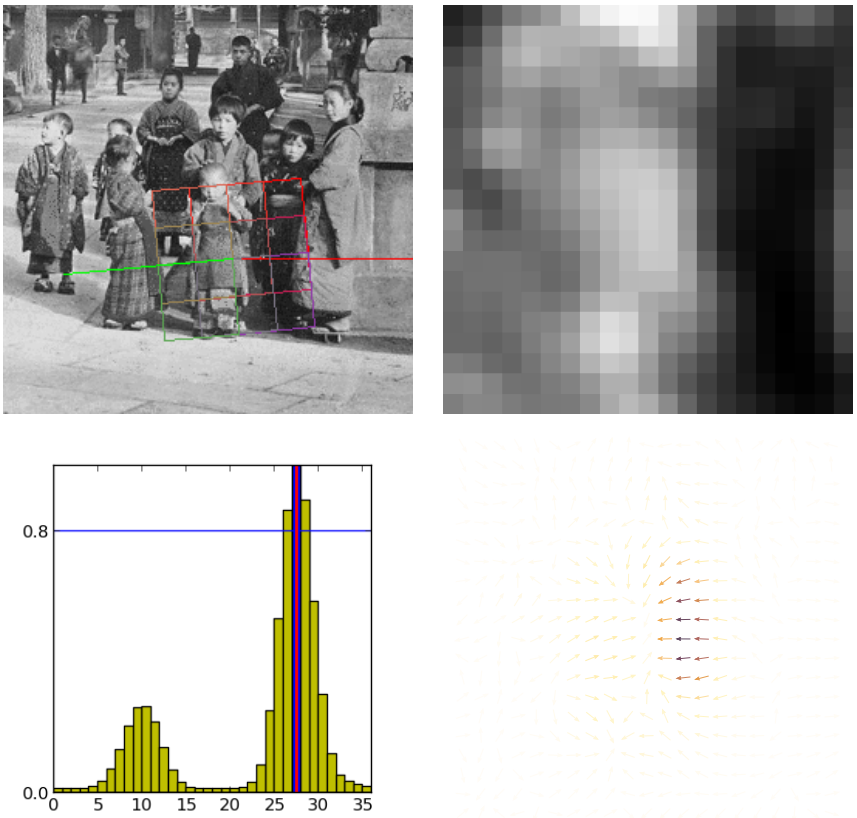


Figure 8: Illustration of the reference orientation attribution. The normalized patch $\mathcal{P}^{\text{ori}}$ (normalized to scale and translation) has a width of $6\lambda_{\text{ori}}\sigma_{\text{key}}$. The gradient magnitude is weighted by a Gaussian window of standard deviation $\lambda_{\text{ori}}\sigma_{\text{key}}$. The gradient orientation are accumulated into an orientation histogram h which is subsequently smoothed.

The normalized patch denoted $\mathcal{P}^{\text{descr}}$ is the set of samples (m,n) of $\mathbf{v}_s^o$ with normalized coordinates $(\hat{x}_{m,n}, \hat{y}_{m,n})$ satisfying

$$\max(|\hat{x}_{m,n}|, |\hat{y}_{m,n}|) \leq \lambda_{\text{descr}}. \tag{4.7}$$

Note that no image re-sampling is performed. Each of these samples (m,n) is characterized by the gradient orientation normalized with respect to the keypoint orientation θ_{key},

$$\hat{\theta}_{m,n} = \text{arctan2}\left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) - \theta_{\text{key}} \bmod 2\pi, \tag{4.8}$$

and its total contribution $c^{\text{descr}}_{m,n}$, which is the product of its gradient norm and a Gaussian weight (with standard deviation $\lambda_{\text{descr}}\sigma_{\text{key}}$) reducing the contribution of distant pixels,

$$c^{\text{descr}}_{m,n} = \frac{1}{\sqrt{2\pi}\lambda_{\text{descr}}\sigma} e^{-\frac{\|(m\delta^o, n\delta^o) - (x,y)\|^2}{2(\lambda_{\text{descr}}\sigma)^2}} \left\| \left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) \right\|. \tag{4.9}$$

The array of orientation histograms. The gradient orientation of each pixel in the normalized patch $\mathcal{P}^{\text{descr}}$ is accumulated into an array of $n_{\text{hist}} \times n_{\text{hist}}$ orientation histograms (standard value $n^{\text{hist}} = 4$). Each of these histograms, denoted $h^{i,j}$ for $(i,j) \in \{1,\dots,n^{\text{hist}}\}^2$, has an associated position with respect to the keypoint $(x_{\text{key}}, y_{\text{key}}, \sigma_{\text{key}}, \theta_{\text{key}})$, given by

$$\hat{x}^i = \left(i - \frac{1+n^{\text{hist}}}{2}\right)\frac{2\lambda_{\text{descr}}}{n^{\text{hist}}} \quad , \quad \hat{y}^j = \left(j - \frac{1+n^{\text{hist}}}{2}\right)\frac{2\lambda_{\text{descr}}}{n^{\text{hist}}}.$$

Each histogram $h^{i,j}$ consists of n^{ori} bins $h^{i,j}_k$ with $k \in \{1,\dots,n^{\text{ori}}\}$, centered on $\hat{\theta}^k = 2\pi(k-1)/n^{\text{ori}}$ (standard value $n^{\text{ori}} = 8$). Each sample (m,n) in the normalized patch $\mathcal{P}^{\text{descr}}$ contributes to the nearest histograms (up to four histograms). Its total contribution $c^{\text{descr}}_{m,n}$ is split bi-linearly over the nearest histograms depending on the distances to each of them (see Figure 10). In the same way, the contribution within each histogram is subsequently split linearly between the two nearest bins. This results, for the sample (m,n), in the following updates. For every $(i,j,k) \in \{1,\dots,n^{\text{hist}}\}^2 \times \{1,\dots,n^{\text{ori}}\}$ such that $|\hat{x}^i - \hat{x}_{m,n}| \leq \frac{2\lambda_{\text{descr}}}{n^{\text{hist}}}$, $|\hat{y}^j - \hat{y}_{m,n}| \leq \frac{2\lambda_{\text{descr}}}{n^{\text{hist}}}$ and $|\hat{\theta}^k - \hat{\theta}_{m,n} \bmod 2\pi| \leq \frac{2\pi}{n^{\text{ori}}}$,

$$h^{i,j}_k \leftarrow h^{i,j}_k + \left(1 - \frac{n^{\text{hist}}}{2\lambda_{\text{descr}}}\left|\hat{x}^i - \hat{x}_{m,n}\right|\right)\left(1 - \frac{n^{\text{hist}}}{2\lambda_{\text{descr}}}\left|\hat{y}^j - \hat{y}_{m,n}\right|\right)\left(1 - \frac{n^{\text{ori}}}{2\pi}\left|\theta^k - \hat{\theta}_{m,n} \bmod 2\pi\right|\right) c^{\text{descr}}_{m,n}. \tag{4.10}$$

The SIFT feature vector. The accumulated array of histograms are encoded into a vector feature $\mathbf{f}$ of length $n^{\text{hist}} \times n^{\text{hist}} \times n^{\text{ori}}$, as follows:

$$\mathbf{f}_{(i-1)n^{\text{hist}}n^{\text{ori}} + (j-1)n^{\text{ori}} + k} = h^{i,j}_k,$$

where $i = 1,\dots,n^{\text{hist}}$, $j = 1,\dots,n^{\text{hist}}$ and $k = 1,\dots,n^{\text{ori}}$. The components of the feature vector $\mathbf{f}$ are saturated to a maximum value of 20% of its Euclidean norm, i.e, $\mathbf{f}_k \leftarrow \min(\mathbf{f}_k, 0.2\|\mathbf{f}\|)$ and then re-normalized to have $\|\mathbf{f}\| = 1$. The saturation of the feature vector components seeks to reduce the impact of non-linear illumination changes, such as saturated regions. The vector is finally renormalized to set the vector maximum value to 255 and finally quantized to 8 bit integers. This is done to accelerate the computation of distances between feature vectors of different images.

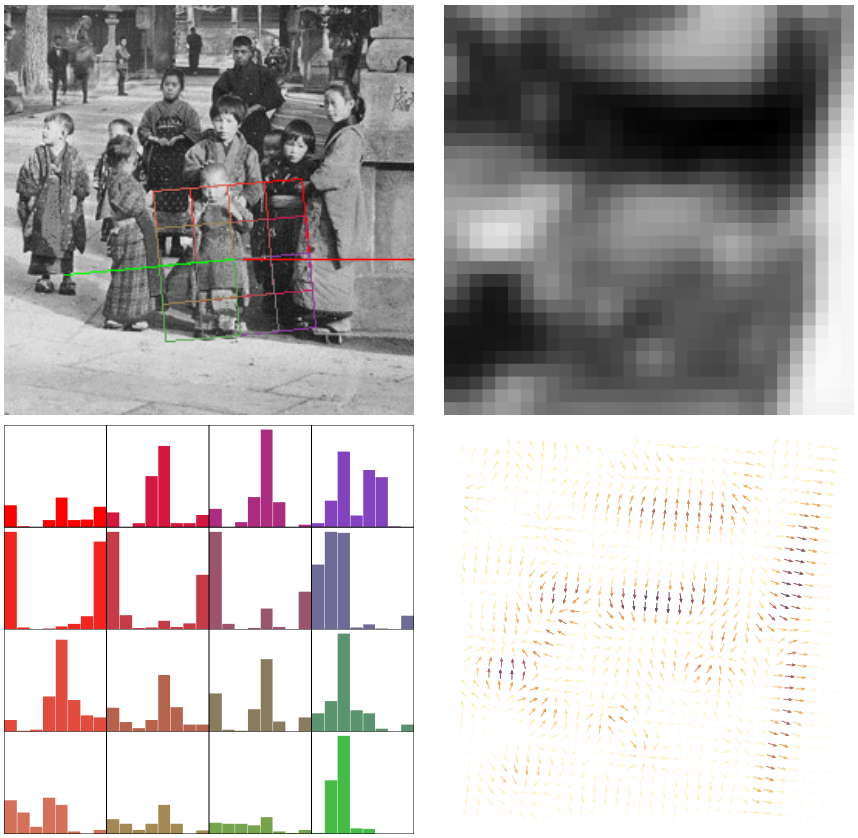


Figure 9: Illustration of the SIFT descriptor construction. No explicit re-sampling of the described normalized patch is performed. The normalized patch $\mathcal{P}^{\text{descr}}$ is partitioned into a set of $n^{\text{hist}} \times n^{\text{hist}}$ subpatches (with here $n^{\text{hist}} = 4$). Each sample (m, n) inside $\mathcal{P}^{\text{descr}}$ (located at $(m\delta^o, n\delta^o)$) contributes by an amount which is a function of their normalized coordinates $(\hat{x}_{m,n}, \hat{y}_{m,n})$ (see (4.6)). Each sub-patch $\mathcal{P}^{\text{descr}}_{(i,j)}$ is centered at $(\hat{x}_i, \hat{y}_j)$.

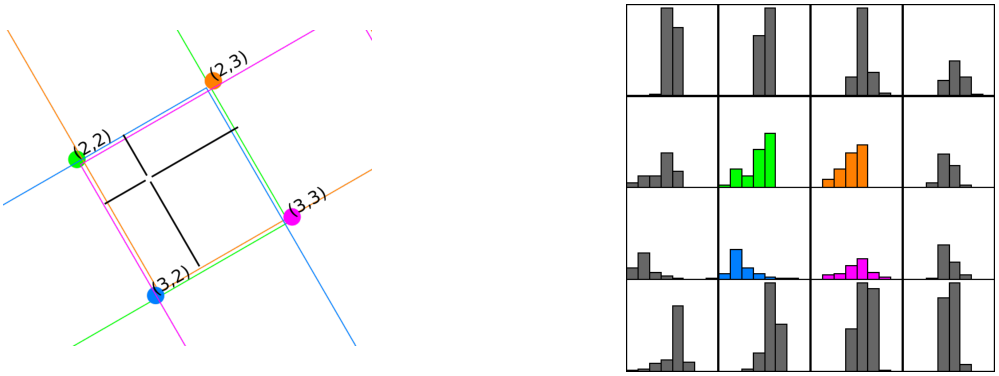


Figure 10: Illustration of the bi-linear spatial sharing of the contribution of a sample inside the patch $\mathcal{P}^{\text{descr}}$. The sample (m, n) contributes to the weighted histograms $(2, 2)$ (green), $(2, 3)$ (orange), $(3, 2)$ (blue) and $(3, 3)$ (pink); The contribution $c^{\text{descr}}_{m,n}$ is split over four pairs of bins according to (4.10).

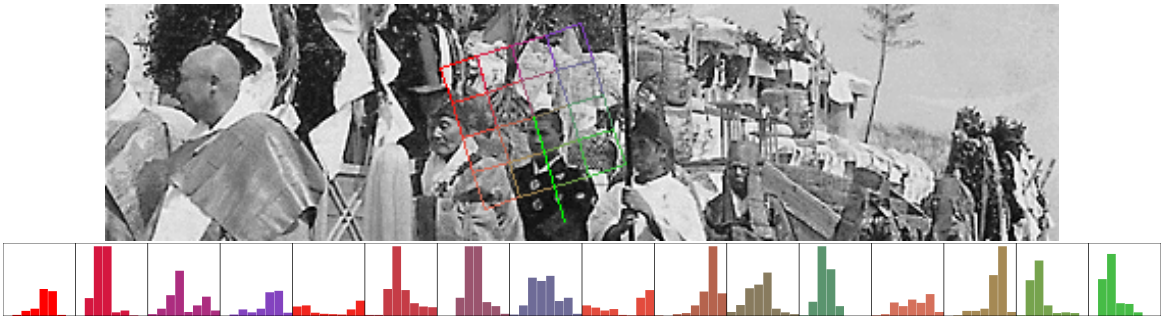


Figure 11: Array of histograms corresponding to an example keypoint is converted into a vector, that undergoes a threshold and quantization. The first picture features the $n^{\text{hist}} \times n^{\text{hist}}$ array sub-patches relative to a keypoint; the corresponding n^{ori} bins histograms are rearranged into a 1D-vector $\vec{v}$. This vector is subsequently thresholded and normalized so its Euclidean norm is 1. The dimension of the feature vector in this example is 128, relative to parameter $n^{\text{hist}} = 4$, $n^{\text{ori}} = 8$ (standard values).

4.3 Pseudocodes

Algorithm 10: Computation of the 2D gradient at each image of the scale-space

Input: $(\mathbf{v}_s^o)$, digital Gaussian scale-space, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n_{\text{spo}} + 2$.

Outputs: - $(\partial_m \mathbf{v}^o_{s,m,n})$, scale-space gradient along x, $o = 1, \ldots, n_{\text{oct}}$ and $s = 1, \ldots, n_{\text{spo}}$.
- $(\partial_n \mathbf{v}^o_{s,m,n})$, scale-space gradient along y, $o = 1, \ldots, n_{\text{oct}}$ and $s = 1, \ldots, n_{\text{spo}}$.

for $o = 1, \ldots, n_{oct}$ **and** $s = 1, \ldots, n_{spo}$ **do**
  for $m = 1, \ldots, M_o - 2$ **and** $n = 1, \ldots, N_o - 2$ **do**
    $\partial_m \mathbf{v}^o_{s,m,n} = (\mathbf{v}^o_{s,m+1,n} - \mathbf{v}^o_{s,m-1,n})/2$
    $\partial_n \mathbf{v}^o_{s,m,n} = (\mathbf{v}^o_{s,m,n+1} - \mathbf{v}^o_{s,m,n-1})/2$

Algorithm 11: Computing the keypoint reference orientation

Inputs: - $(\partial_m \mathbf{v}^o_{s,m,n})$, scale-space gradient along x, $o = 1, \dots, n_{\text{oct}}$ and $s = 1, \dots, n_{\text{spo}}$.
- $(\partial_n \mathbf{v}^o_{s,m,n})$, scale-space gradient along y, $o = 1, \dots, n_{\text{oct}}$ and $s = 1, \dots, n_{\text{spo}}$.
- $\mathcal{L}_{\text{C}} = \{(o_{\text{key}}, s_{\text{key}}, x_{\text{key}}, y_{\text{key}}, \sigma_{\text{key}}, \boldsymbol{\omega})\}$, list of keypoints.

Parameters: - λ_{ori}. The patch $\mathcal{P}^{\text{ori}}$ is $6\lambda_{\text{ori}}\sigma$ wide.
The Gaussian window has a standard deviation of $\lambda_{\text{ori}}\sigma$.
- n^{bins}, number of bins in the orientation histogram h.
- t, threshold for secondary reference orientations.

Output: $\mathcal{L}_{\text{D}} = \{(o, s', m', n', x, y, \sigma, \boldsymbol{\omega}, \theta)\}$ list of oriented keypoints.

Temporary: h_k, orientation histogram, $k = 1, \dots, n^{\text{bins}}$ and with h_k covering $[\frac{2\pi(k-3/2)}{n^{\text{bins}}}; \frac{2\pi(k-1/2)}{n^{\text{bins}}}]$.

for each *keypoint* $(o_{key}, s_{key}, x_{key}, y_{key}, \sigma_{key}, \boldsymbol{\omega})$ *in* $\mathcal{L}_C$ **do**

`// Initialize the orientation histogram` h

for $1 \leq k \leq n^{bins}$ **do** $h_k \leftarrow 0$

`// Accumulate samples from the normalized patch` $\mathcal{P}^{\text{ori}}$ `(eq.(4.1).`

for $m = [(x_{key} - 3\lambda_{ori}\sigma_{key})/\delta_{o_{key}}], \dots, [(x_{key} + 3\lambda_{ori}\sigma_{key})/\delta_{o_{key}}]$ **do**

for $n = [(y_{key} - 3\lambda_{ori}\sigma_{key})/\delta_{o_{key}}], \dots, [(y_{key} + 3\lambda_{ori}\sigma_{key})/\delta_{o_{key}}]$ **do**

`// Compute the sample contribution`

$$c^{\text{ori}}_{m,n} = \frac{1}{\sqrt{2\pi}\lambda_{\text{ori}}\sigma_{\text{key}}} e^{-\frac{\|(m\delta_{o_{\text{key}}}, n\delta_{o_{\text{key}}}) - (x_{\text{key}}, y_{\text{key}})\|^2}{2(\lambda_{\text{ori}}\sigma_{\text{key}})^2}} \left\| \left(\partial_m \mathbf{v}^{o_{\text{key}}}_{s_{\text{key}},m,n}, \partial_n \mathbf{v}^{o_{\text{key}}}_{s_{\text{key}},m,n}\right) \right\|$$

`// Compute the corresponding bin index`

$$b^{\text{ori}}_{m,n} = \left\lceil \frac{n^{\text{bins}}}{2\pi} \left(\arctan2\left(\partial_m \mathbf{v}^{o_{\text{key}}}_{s_{\text{key}},m,n}, \partial_n \mathbf{v}^{o_{\text{key}}}_{s_{\text{key}},m,n}\right) \bmod 2\pi \right) \right\rceil$$

`// Update the histogram`

$$h_{b^{\text{ori}}_{m,n}} \leftarrow h_{b^{\text{ori}}_{m,n}} + c^{\text{ori}}_{m,n}$$

`// Smooth` h

Apply six times a circular convolution with filter $[1, 1, 1]/3$ to h.

`// Extract the reference orientations`

for $1 \leq k \leq n^{bins}$ **do**

if $h_k > h_{k^-}$, $h_k > h_{k^+}$ *and* $h_k \geq t \max(h)$ **then**

`// Compute the reference orientation` θ_{key}

$$\theta_{\text{key}} = \theta_k + \frac{\pi}{n^{\text{bins}}} \left(\frac{h_{k^-} - h_{k^+}}{h_{k^-} + 2h_k + h_{k^+}} \right)$$

note: $[\cdot]$ denotes the round function.

Algorithm 12: Construction of the keypoint descriptor

Inputs: - $(\partial_m \mathbf{v}^o_{s,m,n})$, scale-space gradient along x.
- $(\partial_n \mathbf{v}^o_{s,m,n})$, scale-space gradient along y (see Algorithm 10).
- $\mathcal{L}_{\mathrm{D}} = \{(o_{\mathrm{key}}, s_{\mathrm{key}}, x_{\mathrm{key}}, y_{\mathrm{key}}, \sigma_{\mathrm{key}}, \theta_{\mathrm{key}})\}$ list of keypoints.

Output: $\mathcal{L}_{\mathrm{E}} = \{(o_{\mathrm{key}}, s_{\mathrm{key}}, x_{\mathrm{key}}, y_{\mathrm{key}}, \sigma_{\mathrm{key}}, \theta_{\mathrm{key}}, \mathbf{f})\}$ list of keypoints with feature vector $\mathbf{f}$.

Parameters: - n^{hist}. The descriptor is an array of $n^{\mathrm{hist}} \times n^{\mathrm{hist}}$ orientation histograms.
- n^{ori}, number of bins in the orientation histograms.
Feature vectors $\mathbf{f}$ have a length of $n^{\mathrm{hist}} \times n^{\mathrm{hist}} \times n^{\mathrm{ori}}$
- λ_{descr}.
The Gaussian window has a standard deviation of $\lambda_{\mathrm{descr}}\sigma_{\mathrm{key}}$.
The patch $\mathcal{P}^{\mathrm{descr}}$ is $2\lambda_{\mathrm{descr}}\frac{n^{\mathrm{hist}}+1}{n^{\mathrm{hist}}}\sigma_{\mathrm{key}}$ wide.

Temporary: $h_k^{i,j}$, array of orientation weighted histograms, $(i,j) \in \{1,\dots,n^{\mathrm{hist}}\}$ and $k \in \{1,\dots,n^{\mathrm{ori}}\}$

for each *keypoint* $(o_{key}, s_{key}, x_{key}, y_{key}, \sigma_{key}, \theta_{key})$ *in* $\mathcal{L}_D$ **do**

`// Initialize the array of weighted histograms`
for $1 \le i \le n^{hist}$, $1 \le j \le n^{hist}$ **and** $1 \le k \le n^{ori}$ **do** $h_k^{i,j} \leftarrow 0$

`// Accumulate samples of normalized patch` $\mathcal{P}^{\mathtt{descr}}$ `in the array histograms (eq.(4.7))`
for $m = \left[\left(x_{key} - \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n^{hist}+1}{n^{hist}}\right)/\delta_o\right], \dots, \left[\left(x_{key} + \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n^{hist}+1}{n^{hist}}\right)/\delta_o\right]$ **do**
for $n = \left[\left(y_{key} - \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n^{hist}+1}{n^{hist}}\right)/\delta_o\right], \dots, \left[\left(y_{key} + \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n^{hist}+1}{n^{hist}}\right)/\delta_o\right]$ **do**

`// Compute normalized coordinates (eq.(4.6)).`
$\hat{x}_{m,n} = \left((m\delta_{o_{\mathrm{key}}} - x_{\mathrm{key}})\cos\theta_{\mathrm{key}} + (n\delta_{o_{\mathrm{key}}} - y_{\mathrm{key}})\sin\theta_{\mathrm{key}}\right)/\sigma_{\mathrm{key}}$
$\hat{y}_{m,n} = \left(-(m\delta_{o_{\mathrm{key}}} - x_{\mathrm{key}})\sin\theta_{\mathrm{key}} + (n\delta_{o_{\mathrm{key}}} - y_{\mathrm{key}})\cos\theta_{\mathrm{key}}\right)/\sigma_{\mathrm{key}}$

`// Verify if the sample` (m,n) `is inside the normalized patch` $\mathcal{P}^{\mathtt{descr}}$.
if $\max(|\hat{x}_{m,n}|, |\hat{y}_{m,n}|) < \lambda_{descr}\frac{n^{hist}+1}{n^{hist}}$ **then**

`// Compute normalized gradient orientation.`
$\hat{\theta}_{m,n} = \arctan2\left(\partial_m \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}, \partial_n \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}\right) - \theta_{\mathrm{key}} \bmod 2\pi$

`// Compute the total contribution of the sample` (m,n)
$c^{\mathrm{descr}}_{m,n} = \frac{1}{\sqrt{2\pi}\lambda_{\mathrm{descr}}\sigma_{\mathrm{key}}} e^{-\frac{\|(m\delta^{o_{\mathrm{key}}}, n\delta^{o_{\mathrm{key}}}) - (x_{\mathrm{key}}, y_{\mathrm{key}})\|^2}{2(\lambda_{\mathrm{descr}}\sigma_{\mathrm{key}})^2}} \left\|\left(\partial_m \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}, \partial_n \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}\right)\right\|$

`// Update the nearest histograms and the nearest bins (eq.(4.10)).`
for $(i,j) \in \{1,\dots,n^{hist}\}^2$ *such that* $|\hat{x}^i - \hat{x}_{m,n}| \le \frac{2\lambda_{descr}}{n^{hist}}$ *and* $|\hat{y}^j - \hat{y}_{m,n}| \le \frac{2\lambda_{descr}}{n^{hist}}$ **do**
for $k \in \{1,\dots,n^{ori}\}$ *such that* $\left|\hat{\theta}^k - \hat{\theta}_{m,n} \bmod 2\pi\right| < \frac{2\pi}{n^{ori}}$ **do**
$h_k^{i,j} \leftarrow h_k^{i,j} +$
$\left(1 - \frac{n^{\mathrm{hist}}}{2\lambda_{\mathrm{descr}}}|\hat{x}_{m,n} - \hat{x}^i|\right)\left(1 - \frac{n^{\mathrm{hist}}}{2\lambda_{\mathrm{descr}}}|\hat{y}_{m,n} - \hat{y}^j|\right)\left(1 - \frac{n^{\mathrm{ori}}}{2\pi}|\hat{\theta}_{m,n} - \hat{\theta}^k \bmod 2\pi|\right) c^{\mathrm{descr}}_{m,n}$

`// Build the feature vector` $\mathbf{f}$ `from the array of weighted histograms.`
for $1 \le i \le n^{hist}$, $1 \le j \le n^{hist}$ **and** $1 \le k \le n^{ori}$ **do**
$\mathbf{f}_{(i-1)n^{\mathrm{hist}}n^{\mathrm{ori}} + (j-1)n^{\mathrm{ori}} + k} = h_k^{i,j}$
for $1 \le l \le n^{hist} \times n^{hist} \times n^{ori}$ **do**
$\mathbf{f}_l \leftarrow \min\left(\mathbf{f}_l / \|\mathbf{f}\|, 0.2\right)$ `/*normalize and threshold f*/`
$\mathbf{f}_l \leftarrow \lfloor 256\mathbf{f}_l \rfloor$ `/*quantize to 8 bit integers*/`

Add $(x, y, \sigma, \theta, \mathbf{f})$ to $\mathcal{L}_{\mathrm{E}}$

5 Matching

The classical purpose of detecting and describing keypoints is to find matches (pairs of keypoints) between two images. In the absence of extra knowledge on the problem (in the form of geometric constraints for instance) a matching procedure should consist of two steps: the pairing of similar keypoints from respective images and the selection of those that are reliable. Many algorithms have been proposed to solve this problem efficiently. In what follows, we present a very simple matching method described in the original article by D. Lowe [1]. Let $\mathcal{L}^A$ and $\mathcal{L}^B$ be the set of descriptors associated to the keypoints detected in images $\mathbf{u}^A$ and $\mathbf{u}^B$. The matching is done by considering every descriptor associated to the list $\mathcal{L}^A$ and finding one possible match in list $\mathcal{L}^B$. The first descriptor $\mathbf{f}^a \in \mathcal{L}^A$ is paired to the descriptor $\mathbf{f}^b \in \mathcal{L}^B$ that minimizes the Euclidean distance between descriptors,

$$\mathbf{f}^b = \arg\min_{\mathbf{f} \in \mathcal{L}^B} \|\mathbf{f} - \mathbf{f}^a\|_2.$$

Pairing a keypoint with descriptor $\mathbf{f}^a$ requires then to compute distances to all descriptors in $\mathcal{L}^B$. This pair is considered reliable only if its absolute distance is below a certain threshold $C^{\text{match}}_{\text{absolute}}$. Otherwise it is discarded. The difficulty to setting this threshold constitutes nevertheless a major drawback of this approach. Alternatively, the distance to the second nearest neighbor can be used to define what constitutes a reliable match. For example, by considering an adaptive threshold $\|\mathbf{f}^a - \mathbf{f}^{b'}\| C^{\text{match}}_{\text{relative}}$, where $\mathbf{f}_{b'}$ is the second nearest neighbor

$$\mathbf{f}^{b'} = \arg\min_{\mathbf{f} \in \mathcal{L}^B \setminus \{\mathbf{f}^b\}} \|\mathbf{f} - \mathbf{f}^a\|_2.$$

A description of this algorithm is presented in Algorithm 13. The major drawback of using a relative threshold is that it omits detections for keypoints associated to a repeated structure in the image (indeed, in such situation the distance to the nearest and second nearest descriptor are comparable).

Algorithm 13: Matching keypoints

Inputs: - $\mathcal{L}^A = \{(x^a, y^a, \sigma^a, \theta^a, \mathbf{f}^a)\}$ keypoints and descriptors relative to image $\mathbf{u}^A$.
- $\mathcal{L}^B = \{(x^b, y^b, \sigma^b, \theta^b, \mathbf{f}^b)\}$ keypoints and descriptors relative to image $\mathbf{u}^B$.

Output: $\mathcal{M} = \{(x^a, y^a, \sigma^a, \theta^a, \mathbf{f}^a), (x^b, y^b, \sigma^b, \theta^b, \mathbf{f}^b)\}$ list of matches with positions.

Parameter: $C^{\text{match}}_{\text{relative}}$ relative threshold

for each *descriptor* $\mathbf{f}^a$ *in* $\mathcal{L}^A$ **do**

  Find $\mathbf{f}^b$ and $\mathbf{f}^{b'}$, nearest and second nearest neighbors of $\mathbf{f}^a$:

    for each *descriptor* $\mathbf{f}$ *in* $\mathcal{L}^B$ **do**

      Compute distance $d(\mathbf{f}^a, \mathbf{f})$

  Select pairs satisfying a relative threshold.

    if $d(\mathbf{f}^a, \mathbf{f}^b) < C^{match}_{relative}\, d(\mathbf{f}^a, \mathbf{f}^{b'})$ **then**

      Add pair $(\mathbf{f}^a, \mathbf{f}^b)$ to $\mathcal{M}$

6 Summary of Parameters

The online demo provided with this publication examines in detail the behavior of each stage of the SIFT algorithm. In what follows, we present all the parameters that can be adjusted in the demo and their expected influence on the behavior of the algorithm.

Digital scale-space configuration and keypoints detection

Parameter	Value	Description
n_{oct}	8	Number of octaves (limited by the image size)
n_{spo}	3	Number of scales per octave
σ_{min}	0.8	Blur level of $\mathbf{v}_0^1$ (seed image)
δ_{min}	0.5	The sampling distance in image $\mathbf{v}_0^1$ (corresponds to a $2\times$ interpolation)
σ_{in}	0.5	Assumed blur level in $\mathbf{u}^{\text{in}}$ (input image)
C_{DoG}	0.03	Threshold over the DoG response set for $n_{\text{spo}} = 3$ and the image range in $[0, 1]$
C_{edge}	10	Threshold over the ratio of principal curvatures.

Table 3: Parameters for scale-space discretization and the detection of keypoints

In the present work, the structure of the digital sampling is unequivocally characterized by four structural parameters (n_{oct}, n_{spo}, σ_{min}, δ_{min}) and by the blur level in the input image σ_{in}. The associated online demo allows one to change the value of these parameters. They can be tuned to satisfy specific requirements. For example, by increasing the number of scales per octave n_{spo} and the initial interpolation factor δ_{min} one can increase the precision of the keypoint localization stage. On the other hand, reducing them will result in a faster algorithm.

The image structures that are potentially detected by SIFT have a scale ranging from σ_{min} to $\sigma_{\text{min}} 2^{n_{\text{oct}}}$. Therefore, it may seem natural to choose the lowest possible value of σ_{min} ($\sigma_{\text{min}} = \sigma_{\text{in}}$) and the largest number of octaves allowed by the input image size. However, the relative level of blur (relative to the image sampling grid) in the seed image $\mathbf{v}_0^1$ is $\sigma_{\text{min}}/\delta_{\text{min}}$, resulting in a relative level of blur for image $\mathbf{v}_{n_{\text{spo}}}^o$ of $2\sigma_{\text{min}}/\delta_{\text{min}}$. To guarantee that $\mathbf{v}_0^{o+1} = S_2\mathbf{v}_{n_{\text{spo}}}^o$ (see Section 2) is aliasing free, $\sigma_{\text{min}}/\delta_{\text{min}}$ should be larger than 0.8 [6]. The standard parameter value $\sigma_{\text{min}}/\delta_{\text{min}} = 1.6$ conservatively guarantees an aliasing free scale-space construction.

The threshold on the DoG value C_{DoG} for discarding detections due to noise is undoubtedly the most critical parameter in the detection phase. Unfortunately, since this threshold is closely related to the level of noise in the input image, no universal value can be set. Additionally, the image contrast of the input image plays the inverse role of the noise level. Hence, the threshold C_{DoG} should be set depending on the signal to noise ratio of the input image. Since the DoG approximates $(2^{1/n_{\text{spo}}} - 1)\sigma^2\Delta v$, the threshold C_{DoG} depends on the number of scales per octave n_{spo}.

The threshold C_{edge}, applied to discard keypoints laying on edges, has in practice a negligible impact on the algorithm performance. Indeed, keypoints laying on edges have a large edge response and thus are easily discarded. Nevertheless, image noise may deteriorate the performance since the edge response will be biased.

Computation of the SIFT descriptor

The provided demo allows shows the computation of the keypoint reference orientation, and also the construction of the feature vector for any detected keypoint.

Parameter	Value	Description
n_{bins}	36	Number of bins in the gradient orientation histogram
λ_{ori}	1.5	Sets how local the analysis of the gradient distribution is: - Gaussian window of standard deviation $\lambda_{\text{ori}}\sigma$ - Patch width $6\lambda_{\text{ori}}\sigma$
t	0.80	Threshold for considering local maxima in the gradient orientation histogram
n_{hist}	4	Number of histograms in the normalized patch is $(n_{\text{hist}} \times n_{\text{hist}})$
n_{ori}	8	Number of bins in the descriptor histograms The feature vectors dimension is $n_{\text{hist}} \times n_{\text{hist}} \times n_{\text{ori}}$
λ_{descr}	6	Sets how local the descriptor is: - Gaussian window of standard deviation $\lambda_{\text{descr}}\sigma$ - Descriptor patch width $(n_{\text{hist}}+1)/n_{\text{hist}}2\lambda_{\text{descr}}\sigma$

Table 4: Parameters related to the computation of the keypoint reference orientation and feature vector

The parameter λ_{ori} controls how local the computation of the reference orientation is. Localizing the gradient analysis may result in an increase in the number of orientation references. Indeed, the orientation histogram coming from an isotropic structure is almost flat and has many local maxima. Another parameter of the algorithm, not included in Table 5 because of its insignificant impact, is the level of smoothing applied to the histogram ($N_{\text{conv}} = 6$).

The size of the normalized patch used for computing the SIFT descriptor is governed by λ_{descr}. A larger patch will produce a more discriminative descriptor but will be less robust to complex deformations on the scene. In the same fashion, the number of histograms $n^{\text{hist}} \times n^{\text{hist}}$ and the number of bins n^{ori} can be set to make the feature vector more robust. Accumulating the sample orientation in fewer bins (decreasing n^{ori}) or reducing the number of histograms covering the patch (decreasing n^{hist}) will result in an increase in robustness, at the expense, however, of discriminativity.

Matching of SIFT feature vectors

The SIFT algorithm consists of the detection of the image keypoints and their description. The demo provides additionally two naive algorithms to match SIFT features: an absolute threshold applied on the distance to the nearest keypoint feature or a relative threshold that depends on the distance to the second nearest keypoint feature. An absolute threshold applied on the distance to the nearest keypoint feature is very difficult to set properly. Depending on the matching problem, such absolute threshold can range from 1 to 100 to give acceptable matching results. In a relative threshold matching scenario, increasing the threshold $C^{\text{match}}_{\text{relative}}$ results in an increased number of matches. In particular, pairs corresponding to repeated structures in the image are less likely to be omitted. However this may lead to an increased number of false matches.

Parameter	Value	Description
$C^{\text{match}}_{\text{absolute}}$	1 to 100	Threshold on the distance to the nearest neighbor
$C^{\text{match}}_{\text{relative}}$	0.6	Relative threshold between nearest and second nearest neighbors

Table 5: Parameters of the matching algorithm

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[2] J. Weickert, S. Ishikawa, and A. Imiya, "Linear scale-space has first been proposed in Japan," *Journal of Mathematical Imaging and Vision*, vol. 10, pp. 237–252, 1999.

[3] I. Rey Otero and M. Delbracio, "How to apply Gaussian convolution to images.," *IPOL*.

[4] D. Marr, S. Ullman, and T. Poggio, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. MIT Press, 2010.

[5] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151, 1988.

[6] J. M. Morel and G. Yu, "Is SIFT scale invariant?," *Inverse Problems and Imaging*, vol. 5, pp. 115 – 136, 2011.