IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2014–12–22.
Submitted on 2013–03–27, accepted on 2014–12–02.
ISSN 2105–1232 © 2014 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2014.82

Anatomy of the SIFT Method

Ives Rey-Otero[1], Mauricio Delbracio[2]

[1] CMLA, ENS Cachan, France (ives.rey.otero@gmail.com)
[2] CMLA, ENS Cachan, France and ECE, Duke University (md211@duke.edu)

Communicated by Gabriele Facciolo *Demo edited by* Ives Rey Otero

Abstract

This article presents a detailed description and implementation of the Scale Invariant Feature Transform (SIFT), a popular image matching algorithm. This work contributes to a detailed dissection of SIFT's complex chain of transformations and to a careful presentation of each of its design parameters. A companion online demonstration allows the reader to use SIFT and individually set each parameter to analyze its impact on the algorithm results.

Source Code

The source code (ANSI C), its documentation, and the online demo are accessible at the IPOL web page of this article[1].

Keywords: sift; feature detection; image comparison

1 General Description

The scale invariant feature transform, SIFT [17], extracts a set of descriptors from an image. The extracted descriptors are invariant to image *translation, rotation and scaling (zoom-out)*. SIFT descriptors have also proved to be robust to a wide family of image transformations, such as slight changes of viewpoint, noise, blur, contrast changes, scene deformation, while remaining discriminative enough for matching purposes.

The seminal paper introducing SIFT in 1999 [16] has sparked an explosion of competitors. SURF [2], Harris and Hessian based detectors [19], MOPS [3], ASIFT [28] and SFOP [8], with methods using binary descriptors such as BRISK [13] and ORB [24], are just a few of the successful variants. These add to the numerous non multi-scale detectors such as the Harris-Stephens detector [10], SUSAN [25], the Förstner detector [7], the morphological corner detector [1] and the machine learning based FAST [23] and AGAST [18].

The SIFT algorithm consists of two successive and independent operations: the detection of interesting points (i.e. keypoints) and the extraction of a descriptor associated to each of them. Since these descriptors are robust, they are usually used for matching pairs of images. Object recognition and video stabilization are other popular application examples. Although the descriptors comparison is not strictly speaking a step of the SIFT method, we have included it in our description for the sake of completeness.

[1] https://doi.org/10.5201/ipol.2014.82

The algorithm principle. SIFT detects a series of keypoints from a multiscale image representation. This multiscale representation consists of a family of increasingly blurred images. Each keypoint is a blob-like structure whose center position (x, y) and characteristic scale σ are accurately located. SIFT computes the dominant orientation θ over a region surrounding each one of these keypoints. For each keypoint, the quadruple (x, y, σ, θ) defines the center, size and orientation of a normalized patch where the SIFT descriptor is computed. As a result of this normalization, SIFT keypoint descriptors are in theory invariant to any translation, rotation and scale change. The descriptor encodes the spatial gradient distribution around a keypoint by a 128-dimensional vector. This feature vector is generally used to match keypoints extracted from different images.

The algorithmic chain. In order to attain scale invariance, SIFT is built on the Gaussian scale-space, a multiscale image representation simulating the family of all possible zoom-outs through increasingly blurred versions of the input image (see [27] for a gentle introduction to the subject). In this popular multiscale framework, the Gaussian convolution acts as an approximation of the optical blur, and the Gaussian kernel approximates the camera's point spread function. Thus, the Gaussian scale-space can be interpreted as a family of images, each of them corresponding to a different zoom factor. The Gaussian scale-space representation is presented in Section 2.

To attain translation, rotation and scale invariance, the extracted keypoints must be related to structures that are unambiguously located, both in scale and position. This excludes image corners and edges since they cannot be precisely localized both in scale and space. Image blobs or more complex local structures characterized by their position and size, are therefore the most suitable structures for SIFT.

Detecting and locating keypoints consists in computing 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 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 to each keypoint a reference orientation. This reference is computed from the gradient orientation over a keypoint neighborhood. This step 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 descriptor is described in Section 4.2. This ends the algorithmic chain defining the SIFT algorithm. Additionally, Section 5 illustrates how SIFT descriptors can be used to find local matches between pairs of images. The method presented here is the matching procedure described in the original paper by D. Lowe [16].

This complex chain of transformations is governed by a large number of design parameters. Section 6 summarizes them and provides an analysis of their respective influence. Table 1 presents the details of the adopted notation while the consecutive steps of the SIFT algorithm are summarized in Table 2.

2 The Gaussian Scale-Space

The Gaussian scale-space representation is a family of increasingly blurred images. This blurring process simulates the loss of detail produced when a scene is photographed from farther and farther

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, list of octaves $\mathbf{v} = (\mathbf{v}^o)$, $o = 1, \dots, n_{\text{oct}}$ Each octave $\mathbf{v}^o$ 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), list of octaves $\mathbf{w} = (\mathbf{w}^o)$, $o = 1, \dots, n_{\text{oct}}$ Each octave $\mathbf{w}^o$ is defined on a discrete grid $(s, m, n) \in \{0, \dots, n_{\text{spo}}+1\} \times \{0, \dots, M_o-1\} \times \{0, \dots, N_o-1\}$
$\boldsymbol{\omega}$	DoG value after 3D extremum subpixel refinement
$\partial_x v$	Scale-space gradient along x ($\partial_y v$ along y), defined on continuous domain $(\sigma, \mathbf{x}) \in \mathbb{R}^+ \times \mathbb{R}^2$
$\partial_m \mathbf{v}$	Digital scale-space gradient along x ($\partial_n \mathbf{v}$ along y), list of octaves $(\partial_m \mathbf{v} = (\partial_m \mathbf{v}^o)$, $o = 1, \dots, n_{\text{oct}})$ Each octave $\partial_m \mathbf{v}^o$ is defined on a discrete grid $(s, m, n) \in \{2, \dots, n_{\text{spo}}\} \times \{1, \dots, M_o-2\} \times \{1, \dots, N_o-2\}$
G_ρ	Continuous Gaussian convolution of standard deviation ρ
$\mathbf{G}_\rho$	Digital Gaussian convolution of standard deviation ρ (see (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 1: Summary of the notation used in the article.

(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 $\mathbf{x} = (x, y) \in \mathbb{R}^2$. 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,

$$G_{\sigma_2}(G_{\sigma_1} u)(\mathbf{x}) = G_{\sqrt{\sigma_1^2+\sigma_2^2}} u(\mathbf{x}). \tag{1}$$

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

$$v : (\sigma, \mathbf{x}) \mapsto G_\sigma u(\mathbf{x}). \tag{2}$$

In the case of digital images there is some ambiguity on how to define a discrete counterpart to the continuous Gaussian smoothing operator [9, 22]. In the present work as in Lowe's original work, the digital Gaussian smoothing is implemented as a discrete convolution with samples of a truncated Gaussian kernel.

Stage	Description
1.	Compute the Gaussian scale-space in: $\mathbf{u}$ image out:$\mathbf{v}$ scale-space
2.	Compute the Difference of Gaussians (DoG) in: $\mathbf{v}$ scale-space out: $\mathbf{w}$ DoG
3.	Find candidate keypoints (3D discrete extrema of DoG) in: $\mathbf{w}$ 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: $\mathbf{w}$ 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: $\mathbf{w}$ DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
6.	Filter unstable keypoints laying on edges in: $\mathbf{w}$ DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
7.	Assign a reference orientation to each keypoint in: $(\partial_m \mathbf{v}, \partial_n \mathbf{v})$ scale-space gradient and $\{(x, y, \sigma)\}$ list of keypoints out: $\{(x, y, \sigma, \theta)\}$ list of oriented keypoints
8.	Build the keypoints descriptor in: $(\partial_m \mathbf{v}, \partial_n \mathbf{v})$ scale-space gradient and $\{(x, y, \sigma, \theta)\}$ list of keypoints out: $\{(x, y, \sigma, \theta, \mathbf{f})\}$ list of described keypoints

Table 2: Summary of the SIFT algorithm.

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

$$\mathbf{g}_\sigma(k) = K e^{-\frac{k^2}{2\sigma^2}}, \quad -\lceil 4\sigma \rceil \leq k \leq \lceil 4\sigma \rceil, \quad k \in \mathbb{Z} \tag{3}$$

where $\lceil \cdot \rceil$ denotes the ceil function and K is set so that $\sum \mathbf{g}_\sigma(k) = 1$. 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 by $\mathbf{G}_\sigma \mathbf{u}$, is computed via a separable two-dimensional (2D) discrete convolution

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

where $\bar{\mathbf{u}}$ denotes the extension of $\mathbf{u}$ to $\mathbb{Z}^2$ via symmetrization with respect to $-1/2$, namely,

$$\bar{\mathbf{u}}(k, l) = \mathbf{u}(s_M(k), s_N(l)) \quad \text{with} \quad s_M(k) = \min(k \bmod 2M, 2M - 1 - k \bmod 2M).$$

For the range of values of σ considered in the described algorithm (i.e. $\sigma \geq 0.7$), the digital Gaussian smoothing operator approximately satisfies a semi-group relation with an error below 10^{-4} for pixel intensity values ranging from 0 to 1 [22]. Applying successively two digital Gaussian smoothings of parameters σ_1 and σ_2 is approximately equal to applying one digital Gaussian smoothing of parameter $\sqrt{\sigma_1^2 + \sigma_2^2}$,

$$\mathbf{G}_{\sigma_2}(\mathbf{G}_{\sigma_1} \mathbf{u}) = \mathbf{G}_{\sqrt{\sigma_1^2 + \sigma_2^2}} \mathbf{u}. \tag{5}$$

2.2 Digital Gaussian Scale-Space

As previously introduced, the Gaussian scale-space $v : (\mathbf{x}, \sigma) \mapsto G_\sigma u(\mathbf{x})$ is a family 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 family of digital images relative to a discrete set of blur levels and different sampling rates, all of them derived from an input image $\mathbf{u}_{\text{in}}$ with assumed blur level σ_{in}. This family is split into subfamilies of images sharing a common sampling rate. Since in the original SIFT algorithm the sampling rate is iteratively decreased by a factor of two, these subfamilies are called *octaves*.

Let n_{oct} be the total number of octaves in the digital scale-space, $o \in \{1, \dots, n_{\text{oct}}\}$ 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 default value is $n_{\text{spo}} = 3$). Each octave o contains the images $\mathbf{v}_s^o$ for $s = 1, \dots, n_{\text{spo}}$, each of them with a different blur level σ_s^o. The blur level in the digital scale-space is measured taking as unit length 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.

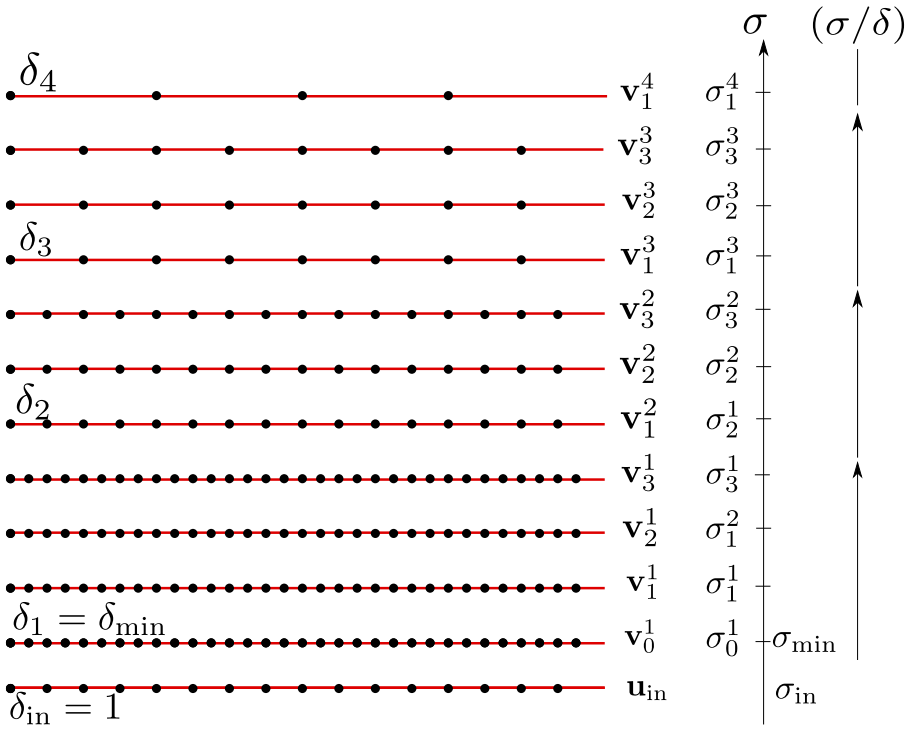


Figure 1: Convention adopted for the sampling grid of the digital scale-space $\mathbf{v}$. The blur level is considered with respect to the sampling grid of the input image. The parameters are set to their default value, namely $\sigma_{\min} = 0.8, n_{\text{spo}} = 3, n_{\text{oct}} = 8, \sigma_{\text{in}} = 0.5$.

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

The construction of the digital scale-space begins with the computation of a *seed* image denoted by $\mathbf{v}_0^1$. This image will have a blur level $\sigma_0^1 = \sigma_{\min}$, which is the minimum blur level considered, and

inter pixel distance $\delta_0 = \delta_{\min}$. It is computed from $\mathbf{u}_{\text{in}}$ by

$$\mathbf{v}_0^1 = \mathbf{G}_{\frac{1}{\delta_{\min}}\sqrt{\sigma_{\min}^2 - \sigma_{\text{in}}^2}} \mathbf{I}_{\delta_{\min}} \mathbf{u}_{\text{in}}, \tag{6}$$

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 default 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 set as default 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

$$\mathbf{v}_s^o = \mathbf{G}_{\rho_{[(s-1)\to s]}} \mathbf{v}_{s-1}^o, \tag{7}$$

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

$$\mathbf{v}_0^o = \mathbf{S}_2 \mathbf{v}_{n_{\text{spo}}}^{o-1}, \tag{8}$$

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 family of images $(\mathbf{v}_s^o)$, $o = 1, \ldots, n_{\text{oct}}$ and $s = 0, \ldots, n_{\text{spo}} + 2$, having inter-pixel distance

$$\delta_o = \delta_{\min} 2^{o-1} \tag{9}$$

and blur level

$$\sigma_s^o = \frac{\delta_o}{\delta_{\min}} \sigma_{\min} 2^{s/n_{\text{spo}}}. \tag{10}$$

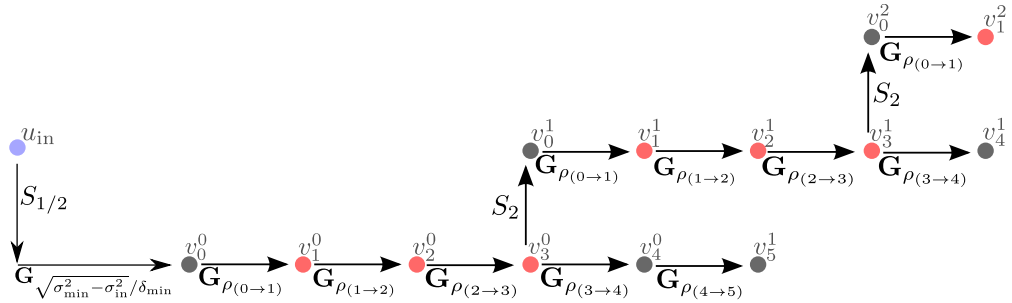
Consequently, the simulated blurs follow a geometric progression. The scale-space construction process is summarized in Algorithm 1. The digital scale-space construction is thus defined by five parameters:

- 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 blur level $\sigma_{\min}$ of the first image of the scale-space $\mathbf{v}_0^1$, and
- σ_{in} the assumed blur level in the input image $\mathbf{u}^{\text{in}}$.

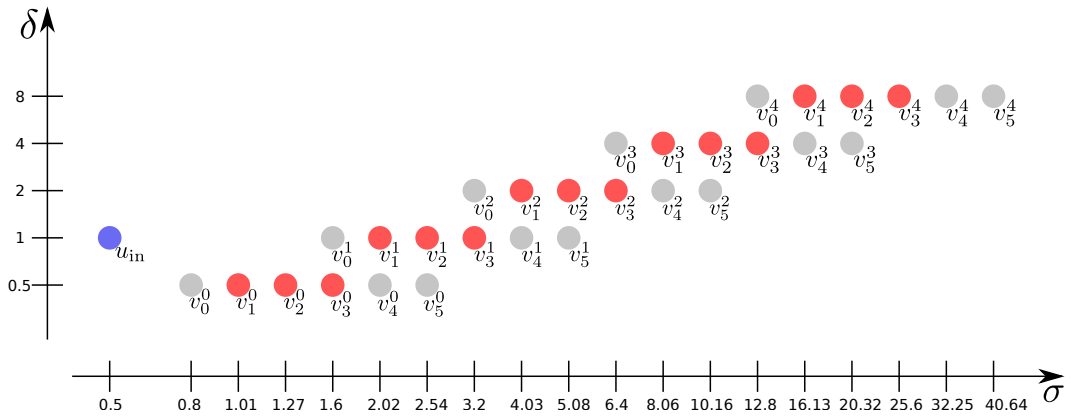
The diagram in Figure 2 depicts the digital scale-space architecture in terms of the sampling rates and blur levels. Each point symbolizes a scale-space image $\mathbf{v}_s^o$ having inter-pixel distance δ^o and blur level σ_s^o. The featured configuration is produced from the default parameter values 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 upper limited by the number of possible subsamplings. Figure 3 shows an example of the digital scale-space images generated with the given configuration.

3 Keypoint Definition

Precisely detecting interesting image features is a challenging problem. The keypoint features are defined in SIFT as the extrema of the normalized Laplacian scale-space $\sigma^2 \Delta v$ [14]. A Laplacian extremum is unequivocally characterized by its scale-space coordinates $(\sigma, \mathbf{x})$ where $\mathbf{x}$ refers to its



(a) Scale-space construction



(b) Scale-space default configuration

Figure 2: (a) The succession of subsamplings and Gaussian convolutions that results in the SIFT scale-space. The first image at each octave $\mathbf{v}_0^o$ is obtained via subsampling, with the exception of the first octave first image $\mathbf{v}_0^0$ that is generated by a bilinear interpolation followed by a Gaussian convolution. (b) An illustration of the digital scale-space in its default 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 blur level and its inter-pixel distance, respectively noted by σ and δ. The scale-space is split into octaves: 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 blur level of the input image is $\sigma_{\text{in}} = 0.5$.

center spatial position and σ relates to its size (scale). As will be presented in Section 4, the covariance of the extremum $(\sigma, \mathbf{x})$ induces the invariance to translation and scale of its associated descriptor.

Instead of computing the Laplacian of the image scale space, SIFT uses a difference of Gaussians operator (DoG), first introduced by Burt and Adelson [4] and Crowley and Stern [6]. Let v be a Gaussian 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 approximately compute the normalized Laplacian $\sigma^2 \Delta v$. Indeed, from a set of simulated blurs following a geometric progression of ratio κ, the heat equation is

Figure 3: Crops of a subset of images extracted from the Gaussian scale-space of an example image. 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.

approximated by

$$\sigma \Delta v = \frac{\partial v}{\partial \sigma} \approx \frac{v(\kappa\sigma, \mathbf{x}) - v(\sigma, \mathbf{x})}{\kappa\sigma - \sigma} = \frac{w(\sigma, \mathbf{x})}{(\kappa - 1)\sigma}. \tag{11}$$

Thus, we have $w(\sigma, \mathbf{x}) \approx (\kappa - 1)\sigma^2 \Delta v(\sigma, \mathbf{x})$.

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. Thus, the discrete extrema of the digital DoG are first detected and then their positions are refined. The detected points are finally validated to discard possible unstable 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 one 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 $\mathbf{v}$. For each octave $o = 1, \ldots, n_{\text{oct}}$ and for each image $\mathbf{w}_s^o$ with $s = 0, \ldots, n_{\text{spo}} + 1$

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

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 (Equation (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 by $\mathbf{v}_s^o(m, n)$.
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 blur level in the input image.

```
//Compute the first octave
  //Compute the seed image v_0^1
    //1.Interpolate the original image (Bilinear interpolation, see Algorithm 2)
    u' ← bilinear_interpolation(u_in, δ_min)
    // 2.  Blur the interpolated image (Gaussian blur, see Equation (4))
    v_0^1 = G_{(1/δ_min)√(σ_min^2 − σ_in^2)} u'
  // Compute the other images in the first octave
  for s = 1, ..., n_spo + 2 do
      v_s^1 = G_{ρ_[(s−1)→s]} v_{s−1}^1

// Compute subsequent octaves
for o = 2, ..., n_oct do
    // Compute the first image in the octave by subsampling
    for m = 0, ..., M_o − 1 and  n = 0, ..., N_o − 1 do
        v_0^o(m, n) ← v_{n_spo}^{o−1}(2m, 2n)
    // Compute the other images in octave o
    for s = 1, ..., n_spo + 2 do
        v_s^o = G_{ρ_[(s−1)→s]} v_{s−1}^o
```

Algorithm 2: Bilinear interpolation of an image

Input: $\mathbf{u}$, digital image, $M \times N$ pixels. The samples are denoted by $\mathbf{u}(m, n)$.
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**
$x \leftarrow \delta' m'$ $y \leftarrow \delta' n'$

$$\begin{aligned}\mathbf{u}'(m', n') \leftarrow\ & (x - \lfloor x \rfloor)(y - \lfloor y \rfloor)\, \bar{\mathbf{u}}(\lfloor x \rfloor + 1, \lfloor y \rfloor + 1) + (1 + \lfloor x \rfloor - x)(y - \lfloor y \rfloor)\, \bar{\mathbf{u}}(\lfloor x \rfloor, \lfloor y \rfloor + 1) \\ & + (x - \lfloor x \rfloor)(1 + \lfloor y \rfloor - y)\, \bar{\mathbf{u}}(\lfloor x \rfloor + 1, \lfloor y \rfloor) + (1 + \lfloor x \rfloor - x)(1 + \lfloor y \rfloor - y)\, \bar{\mathbf{u}}(\lfloor x \rfloor, \lfloor y \rfloor)\end{aligned}$$

$\bar{\mathbf{u}}$ denotes the extension of $\mathbf{u}$ to $\mathbb{Z}^2$ via symmetrization with 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: $\lfloor \cdot \rfloor$ denotes the floor function.

with $m = 0, \dots, M_o - 1$, $n = 0, \dots, N_o - 1$. The image $\mathbf{w}_s^o$ will be attributed the blur level σ_s^o. This computation is illustrated in Figure 4 and summarized in Algorithm 3. See how, in the digital scale-space, the computation of the auxiliary image $\mathbf{v}_{n_{\text{spo}}+2}^o$ 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 Gaussian scale-space $\mathbf{v}$. Figure 6 shows images of an example of DoG scale-space.

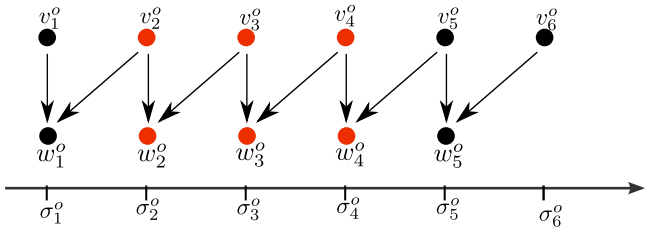


Figure 4: The difference of Gaussians operator is computed by subtracting pairs of contiguous scale-space images. The procedure is not centered: the difference between the images at scales $\kappa\sigma$ and σ is attributed a blur level σ.

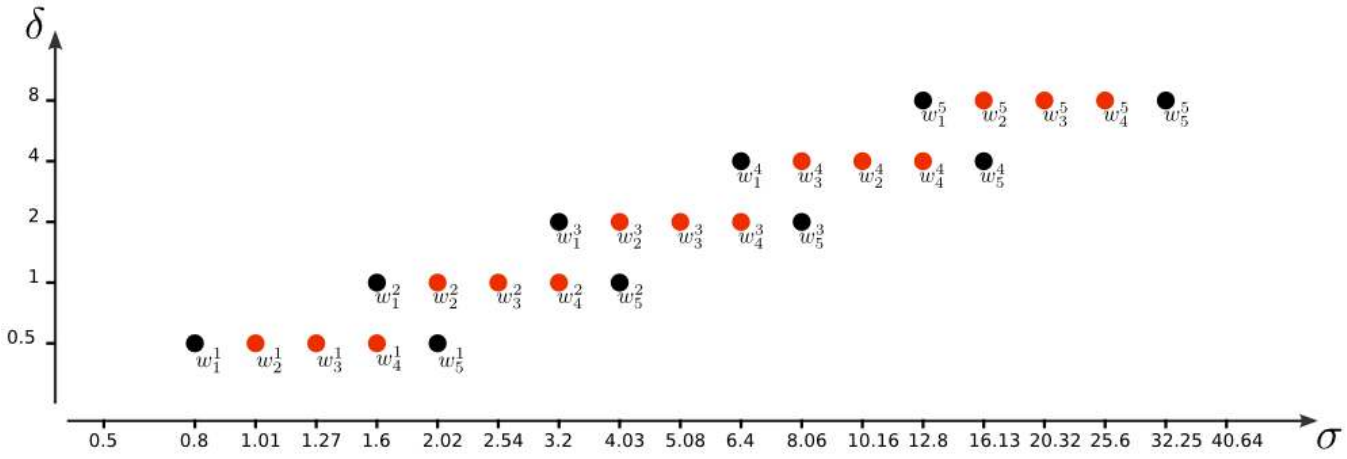


Figure 5: The *DoG* scale-space. The difference of Gaussians acts as an approximation of the normalized Laplacian $\sigma^2\Delta$. The difference $\mathbf{w}_s^o = \mathbf{v}_{s+1}^o - \mathbf{v}_s^o$ is relative to the blur level σ_s^o. Each octave contains n_{spo} images plus two auxiliary images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

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 what follows, 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 $26 = 3 \times 3 \times 3 - 1$). Algorithm 4 summarizes the extraction of 3D extrema from the digital DoG. 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 rudimentary way to detect candidate points. It is sensitive 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. The location of the discrete extrema is constrained to the sampling grid (defined by the octave o). This coarse localization hinders a rigorous covariance property

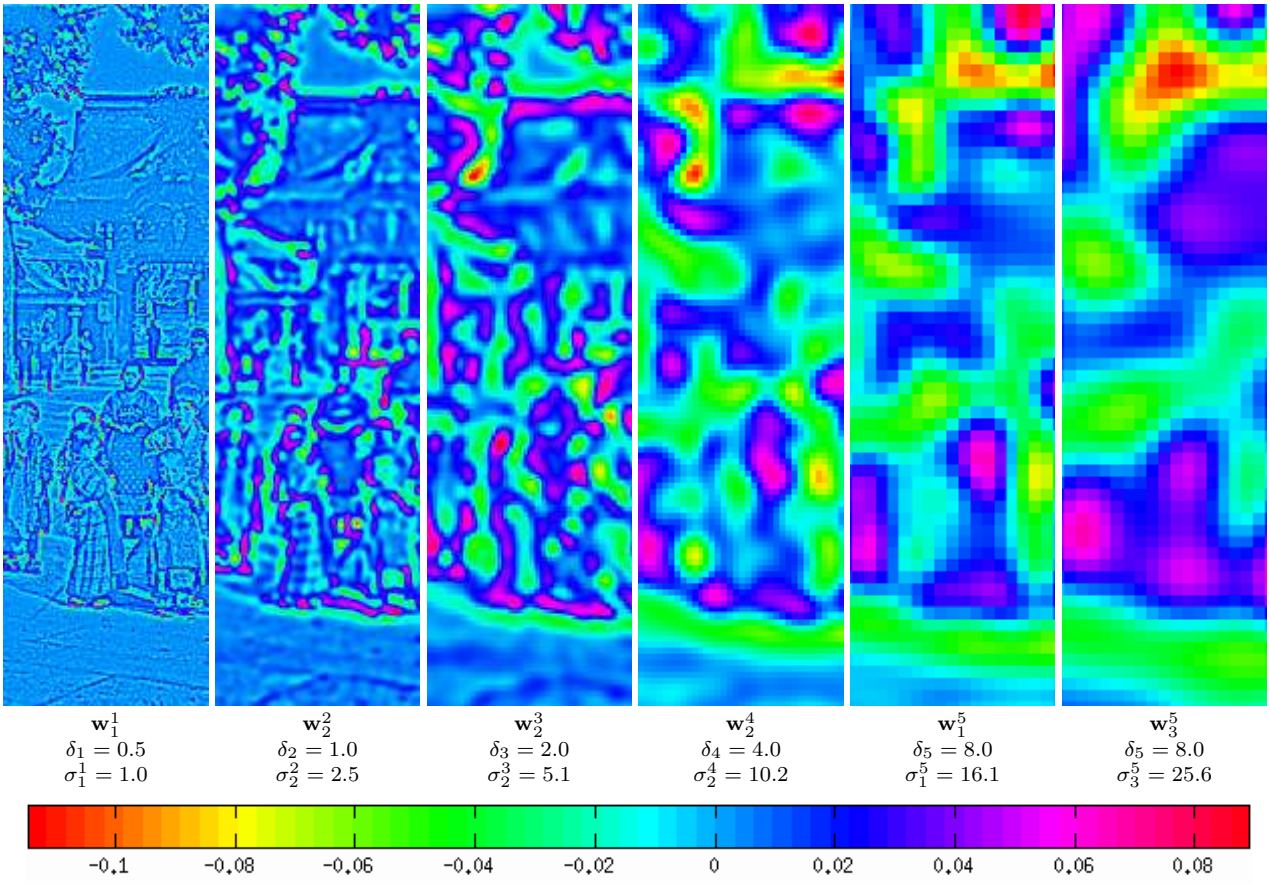


Figure 6: Crops of a subset of images extracted from the DoG scale-space of an example image. The DoG operator is an approximation of the normalized Laplacian operator $\sigma^2 \Delta v$. The DoG scale-space parameters used in this example are the default: $n_{\text{spo}} = 3$, $\sigma_{\min} = 0.8$,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

of the set of keypoints and subsequently is an obstacle to the full scale and translation invariance of the corresponding descriptor. SIFT refines the position and scale of each candidate keypoint using a local interpolation model.

We denote by $\boldsymbol{\omega}^o_{s,m,n}(\boldsymbol{\alpha})$ the quadratic function at sample point (s, m, n) in the octave o, given by

$$\boldsymbol{\omega}^o_{s,m,n}(\boldsymbol{\alpha}) = \mathbf{w}^o_{s,m,n} + \boldsymbol{\alpha}^T \bar{g}^o_{s,m,n} + \frac{1}{2}\boldsymbol{\alpha}^T \bar{H}^o_{s,m,n}\boldsymbol{\alpha}, \tag{12}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^T \in [-1/2, 1/2]^3$; $\bar{g}^o_{s,m,n}$ and $\bar{H}^o_{s,m,n}$ denote respectively the 3D gradient and Hessian at (s, m, n) in the octave o, computed with a finite difference scheme as follows

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

with

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

This quadratic function is 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 SIFT proceeds 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 $\boldsymbol{\omega}^o_{s,m,n}$ by solving $\nabla \boldsymbol{\omega}^o_{s,m,n}(\boldsymbol{\alpha}) = 0$ (see Algorithm 7). This yields
$$\boldsymbol{\alpha}^* = -\left(\bar{H}^o_{s,m,n}\right)^{-1} \bar{g}^o_{s,m,n}. \tag{14}$$
3. If $\max(|\alpha_1^*|, |\alpha_2^*|, |\alpha_3^*|) \leq 0.5$ (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 (10) and (9)), the corresponding keypoint coordinates are
$$(\sigma, x, y) = \left(\frac{\delta_{o_e}}{\delta_{\min}} \sigma_{\min} 2^{(\alpha_1^* + s)/n_{\text{spo}}} \,,\, \delta_{o_e}(\alpha_2^* + m) \,,\, \delta_{o_e}(\alpha_3^* + n) \right). \tag{15}$$
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 (12), the value of the DoG interpolated extremum is

$$\begin{aligned} \boldsymbol{\omega} := \boldsymbol{\omega}^o_{s,m,n}(\boldsymbol{\alpha}^*) &= \mathbf{w}^o_{s,m,n} + (\boldsymbol{\alpha}^*)^T \bar{g}^o_{s,m,n} + \frac{1}{2}(\boldsymbol{\alpha}^*)^T \bar{H}^o_{s,m,n} \boldsymbol{\alpha}^* \\ &= \mathbf{w}^o_{s,m,n} + \frac{1}{2}(\boldsymbol{\alpha}^*)^T \bar{g}^o_{s,m,n}. \end{aligned} \tag{16}$$

This value will be used to discard uncontrasted keypoints.

3.3 Filtering Unstable Keypoints

Discarding Low Contrasted Extrema

Image noise will typically produce several spurious DoG extrema. Such extrema are 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} (see Algorithm 8),

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

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} should depend on n_{spo} (default value $C_{\text{DoG}} = 0.015$ for $n_{\text{spo}} = 3$). The threshold applied in the provided source-code is $\widetilde{C}_{\text{DoG}} = \frac{2^{1/n_{\text{spo}}} - 1}{2^{1/3} - 1} C_{\text{DoG}}$, with C_{DoG} relative to $n_{\text{spo}} = 3$. This guarantees that the applied threshold is independent of the sampling configuration. Before the refinement of the extrema, and to avoid unnecessary computations, a less conservative threshold at 80% of C_{DoG} is applied to the discrete 3D extrema (see Algorithm 5),

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

Discarding candidate keypoints on edges

Candidate keypoints lying on edges are difficult to precisely locate. This is a direct consequence of the fact that an edge is invariant to translations along its principal axis. Such detections do not help define covariant keypoints and should be discarded. The 2D Hessian of the DoG provides a characterization of those undesirable keypoint candidates. 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 difference scheme:

$$H^o_{s,m,n} = \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix}, \tag{17}$$

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

$$\text{edgeness}(H^o_{s,m,n}) = \frac{\text{tr}(H^o_{s,m,n})^2}{\det(H^o_{s,m,n})} = \frac{(\lambda_{\max} + \lambda_{\min})^2}{\lambda_{\max}\lambda_{\min}} = \frac{(r+1)^2}{r}. \tag{18}$$

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 $\bar{H}^o_{s,m,n}$ (13) previously computed for the keypoint interpolation. Algorithm 9 summarizes how keypoints on edges are discarded.

3.4 Pseudocodes

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

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

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

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 Algorithm 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, $o = 1, \dots, n_{\text{oct}}$ and $s = 0, \dots, n_{\text{spo}} + 1$.
- (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 Equation` (13)

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 (17) `// 2D Hessian`
  Compute $\frac{\text{tr}(H^o_{s,m,n})^2}{\det(H^o_{s,m,n})}$ `// edgeness`
  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 Keypoint Description

In the literature, rotation invariant descriptors fall into 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 is based on a normalization. The local dominant gradient angle is computed and used as a reference orientation. Then, the local gradient distribution is normalized with respect to this reference direction (see Figure 7).

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.

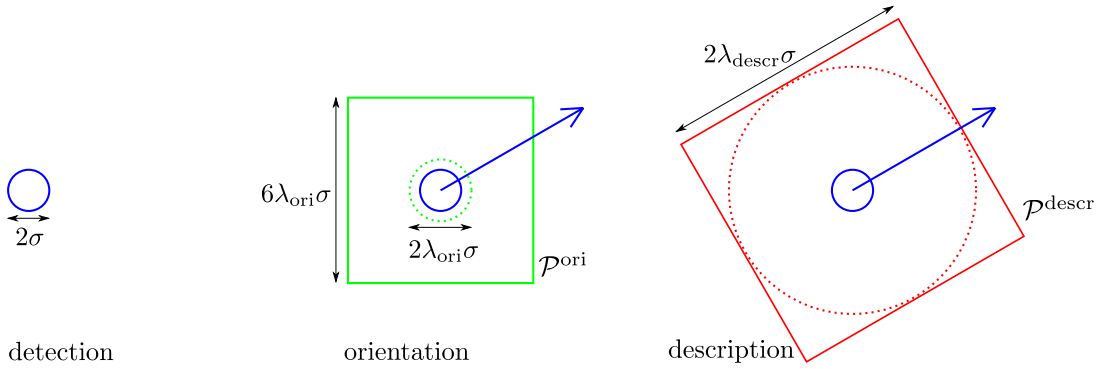


Figure 7: The description of a keypoint detected at scale σ (the radius of the blue circle) involves the analysis of the image gradient distribution around the keypoint in two radial Gaussian neighborhoods with different sizes. **The first local analysis** aims at attributing a reference orientation to the keypoint (the blue arrow). It is performed over a Gaussian window of standard deviation $\lambda_{\text{ori}}\sigma$ (the radius of the green circle). The width of the contributing samples patch $\mathcal{P}^{\text{ori}}$ (green square) is $6\lambda_{\text{ori}}\sigma$. The figure shows the default case $\lambda_{\text{ori}} = 1.5$. **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 approximate width $2\lambda_{\text{descr}}\sigma$. The figure features the default settings: $\lambda_{\text{descr}} = 6$, with a Gaussian window of standard deviation 6σ and a patch $\mathcal{P}^{\text{descr}}$ of width 12σ.

4.1 Keypoint Reference Orientation

A dominant gradient orientation over a keypoint neighborhood is used as its reference orientation. This allows for orientation normalization and hence rotation-invariance of the resulting descriptor (see Figure 7). Computing this reference orientation involves three steps:

A. accumulation of the local distribution of the gradient angle within a normalized patch in an orientation histogram;

B. smoothing of the orientation histogram;

C. extraction of one or more reference orientations from the smoothed histogram.

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

$$\max(|\delta_o m - x|, |\delta_o n - y|) \leq 3\lambda_{\text{ori}}\sigma. \tag{19}$$

Keypoints whose distance to the image borders is less than $3\lambda_{\text{ori}}\sigma$ are discarded since the patch $\mathcal{P}^{\text{ori}}$ is not totally included in the image. 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 $\lambda_{\text{ori}}\sigma$ (default value $\lambda_{\text{ori}} = 1.5$) reducing the contribution of distant pixels.

$$c_{m,n}^{\text{ori}} = e^{-\frac{\|(m\delta_o, n\delta_o) - (x,y)\|^2}{2(\lambda_{\text{ori}}\sigma)^2}} \left\| \left(\partial_m \mathbf{v}_{s,m,n}^o, \partial_n \mathbf{v}_{s,m,n}^o\right) \right\|. \tag{20}$$

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

$$b_{m,n}^{\text{ori}} = \left\lceil \frac{n_{\text{bins}}}{2\pi} \left(\arctan2\left(\partial_m \mathbf{v}_{s,m,n}^o, \partial_n \mathbf{v}_{s,m,n}^o\right)\right) \right\rceil. \tag{21}$$

where $[\cdot]$ denotes the round function and `arctan2` is the two-argument inverse tangent function[2] with range in $[0, 2\pi]$. The gradient components of the scale-space image $\mathbf{v}_o^s$ are computed through a finite difference scheme

$$\partial_m \mathbf{v}^o_{s,m,n} = \frac{1}{2}\left(\mathbf{v}^o_{s,m+1,n} - \mathbf{v}^o_{s,m-1,n}\right), \qquad \partial_n \mathbf{v}^o_{s,m,n} = \frac{1}{2}\left(\mathbf{v}^o_{s,m,n+1} - \mathbf{v}^o_{s,m,n-1}\right), \tag{22}$$

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

B. Smoothing the histogram. After being accumulated, 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). The keypoint reference orientations are taken among the local maxima positions of the smoothed histogram. More precisely, the reference orientations are the positions of local maxima larger than t times the global maximum (default value $t = 0.8$). Let $k \in \{1, \ldots, n_{\text{bins}}\}$ be the index of a bin such that $h_k > h_{k^-}$, $h_k > h_{k^+}$, where $k^- = (k-1) \bmod n_{\text{bins}}$ and $k^+ = (k+1) \bmod n_{\text{bins}}$ and such that $h_k \geq t \max(h)$. This bin is centered at orientation $\theta_k = {}^{2\pi(k-1)}/_{n_{\text{bins}}}$. 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^+},

$$\theta_{\text{ref}} = \theta_k + \frac{\pi}{n_{\text{bins}}}\left(\frac{h_{k^-} - h_{k^+}}{h_{k^-} - 2h_k + h_{k^+}}\right). \tag{23}$$

Each one of the extracted reference orientations leads to the computation of one local descriptor of a keypoint neighborhood. The number of descriptors may consequently exceed the number of keypoints. Figure 8 illustrates how a reference orientation is attibuted to a keypoint.

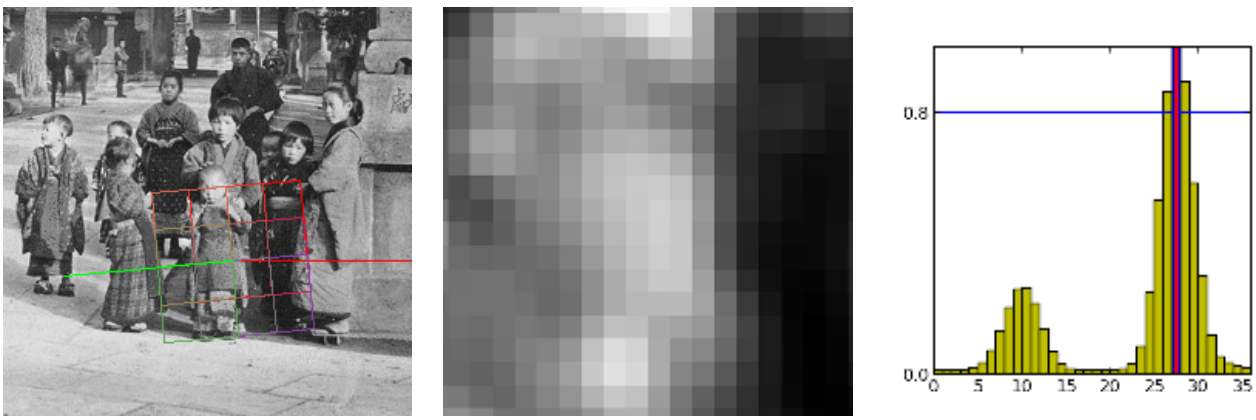


Figure 8: Reference orientation attribution. The normalized patch $\mathcal{P}^{\text{ori}}$ (normalized to scale and translation) width is $6\lambda_{\text{ori}}\sigma_{\text{key}}$. The gradient magnitude is weighted by a Gaussian window of standard deviation $\lambda_{\text{ori}}\sigma_{\text{key}}$.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

[2]The two-argument inverse tangent, unlike the single argument one, determines the appropriate quadrant of the computed angle thanks to the extra information about the signs of the inputs: $\text{arctan2}(x, y) = \arctan({}^x/_y) + \frac{\pi}{2}\text{sign}(y)(1 - \text{sign}(x))$.

4.2 Keypoint Normalized Descriptor

The SIFT descriptor encodes the local spatial distribution of the gradient orientation on a particular neighborhood. SIFT descriptors can be computed anywhere, even densely over the entire image or its scale-space [15, 11]. In the original SIFT method, however, descriptors are computed for all detected keypoints over its normalized neighborhood, making them invariant to translations, rotations and zoom-outs. Given a detected keypoint, the normalized neighborhood consists of a square patch centered at the keypoint and aligned with the reference orientation.

The descriptor consists of a set of weighted histograms of the gradient orientation computed on different regions of the normalized 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 from the Gaussian scale-space image relative to the nearest discrete scale (o, s) to scale σ_{key}, namely $\mathbf{v}_s^o$. A sample (m, n) in $\mathbf{v}_s^o$, 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 $(x_{\text{key}}, y_{\text{key}}, \sigma_{\text{key}}, \theta_{\text{key}})$,

$$\begin{aligned} \hat{x}_{m,n} &= \left((m\delta_o - x_{\text{key}}) \cos\theta_{\text{key}} + (n\delta_o - y_{\text{key}}) \sin\theta_{\text{key}}\right) / \sigma_{\text{key}}, \\ \hat{y}_{m,n} &= \left(-(m\delta_o - x_{\text{key}}) \sin\theta_{\text{key}} + (n\delta_o - y_{\text{key}}) \cos\theta_{\text{key}}\right) / \sigma_{\text{key}}. \end{aligned} \tag{24}$$

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

$$\max(|\hat{x}_{m,n}|, |\hat{y}_{m,n}|) \leq \lambda_{\text{descr}}. \tag{25}$$

Keypoints whose distance to the image borders is less than $\sqrt{2}\lambda_{\text{descr}}\sigma$ are discarded to guaranty that the patch $\mathcal{P}^{\text{descr}}$ is included in the image. Note that no image re-sampling is performed. Each sample (m, n) is characterized by the gradient orientation normalized with respect to the keypoint orientation θ_{key},

$$\hat{\theta}_{m,n} = \operatorname{arctan2}\left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) - \theta_{\text{key}} \bmod 2\pi, \tag{26}$$

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

$$c^{\text{descr}}_{m,n} = e^{-\frac{\|(m\delta^o, n\delta^o) - (x,y)\|^2}{2(\lambda_{\text{descr}}\sigma)^2}} \left\| \left(\partial_m \mathbf{v}^o_{s,m,n}, \partial_n \mathbf{v}^o_{s,m,n}\right) \right\|. \tag{27}$$

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 (default value $n_{\text{hist}} = 4$). Each of these histograms, denoted by $h^{i,j}$ for $(i, j) \in \{1, \ldots, n_{\text{hist}}\}^2$, 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 $k \in \{1, \ldots, n_{\text{ori}}\}$, centered at $\hat{\theta}^k = 2\pi(k-1)/n_{\text{ori}}$ (default 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 bilinearly 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

$$|\hat{x}^i - \hat{x}_{m,n}| \leq \frac{2\lambda_{\text{descr}}}{n_{\text{hist}}}, \quad |\hat{y}^j - \hat{y}_{m,n}| \leq \frac{2\lambda_{\text{descr}}}{n_{\text{hist}}} \quad \text{and} \quad |\hat{\theta}^k - \hat{\theta}_{m,n} \bmod 2\pi| \leq \frac{2\pi}{n_{\text{ori}}},$$

the histogram $h_k^{i,j}$ is updated by

$$h_k^{i,j} \leftarrow h_k^{i,j} + \left(1 - \frac{n_{\text{hist}}}{2\lambda_{\text{descr}}} |\hat{x}^i - \hat{x}_{m,n}|\right) \left(1 - \frac{n_{\text{hist}}}{2\lambda_{\text{descr}}} |\hat{y}^j - \hat{y}_{m,n}|\right) \left(1 - \frac{n_{\text{ori}}}{2\pi} \left|\theta^k - \hat{\theta}_{m,n} \bmod 2\pi\right|\right) c_{m,n}^{\text{descr}}. \tag{28}$$

Figure 9: 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 (default value $n_{\text{hist}} = 4$). Each sample (m,n) inside $\mathcal{P}^{\text{descr}}$, located at $(m\delta^o, n\delta^o)$, contributes by an amount that is a function of their normalized coordinates $(\hat{x}_{m,n}, \hat{y}_{m,n})$ (see (24)). Each sub-patch $\mathcal{P}_{(i,j)}^{\text{descr}}$ is centered at $(\hat{x}_i, \hat{y}_j)$.

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

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

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. $\mathbf{f}_k \leftarrow \min(\mathbf{f}_k, 0.2\|\mathbf{f}\|)$. 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 so as to have $\|\mathbf{f}\|_2 = 512$ and quantized to 8 bit integers as follows: $\mathbf{f}_k \leftarrow \min(\lfloor \mathbf{f}_k \rfloor, 255)$. The quantization aims at accelerating the computation of distances between different feature vectors[3]. Figure 9 and Figure 11 illustrate how a SIFT feature vector is attibuted to an oriented keypoint.

[3] The executable provided by D.Lowe `http://www.cs.ubc.ca/~lowe/keypoints/`, retrieved on September 11th, 2014) uses a different coordinate system (see source code's `README.txt` for details).

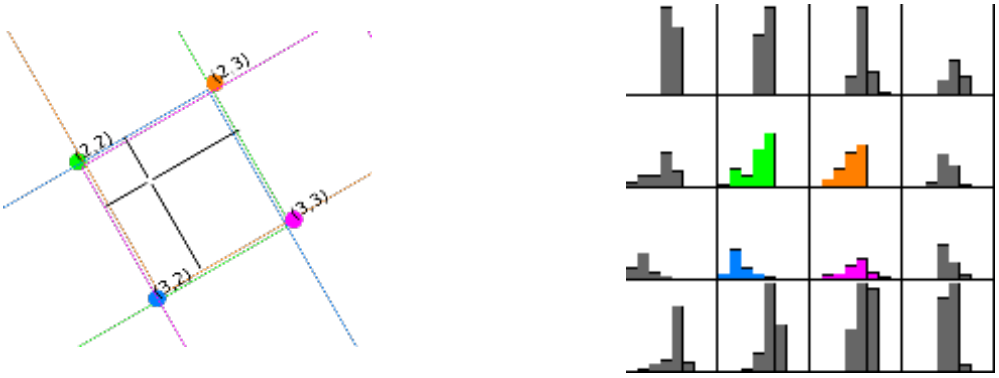


Figure 10: Illustration of the spatial contribution of a sample inside the patch $\mathcal{P}^{\text{descr}}$. The sample (m, n) contributes to the weighted histograms $(2, 2)$ (green), $(2, 3)$ (orange), $(3, 2)$ (blue) and $(3, 3)$ (pink). The contribution $c^{\text{descr}}_{m,n}$ is split over four pairs of bins according to (28).

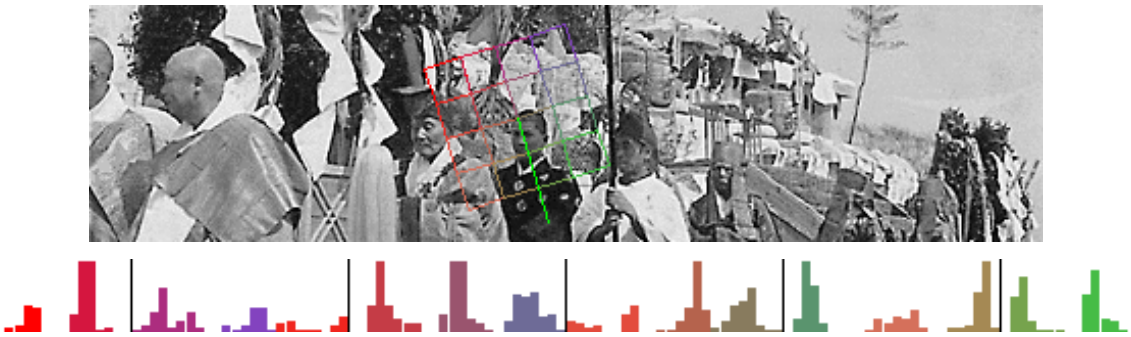


Figure 11: The image on top shows the $n_{\text{hist}} \times n_{\text{hist}}$ array sub-patches relative to a keypoint; the corresponding n_{ori} bins histograms are rearranged into a 1D-vector $\vec{v}$ (bottom). This vector is subsequently thresholded and normalized so that the Euclidean norm equals 512 for each descriptor. The dimension of the feature vector in this example is 128, relative to parameter $n_{\text{hist}} = 4$, $n_{\text{ori}} = 8$ (default values).

4.3 Pseudocodes

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

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

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

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

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

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

`// Check if the keypoint is distant enough from the image borders`

if $3\lambda_{ori}\sigma_{key} \leq x_{key} \leq h - 3\lambda_{ori}\sigma_{key}$ **and** $3\lambda_{ori}\sigma_{key} \leq y_{key} \leq w - 3\lambda_{ori}\sigma_{key}$ **then**

`// Initialize the orientation histogram` h

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

`// Accumulate samples from the normalized patch` $\mathcal{P}^{\mathrm{ori}}$ `(Equation (19).`

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`

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

`// Compute the corresponding bin index`

$$b^{\mathrm{ori}}_{m,n} = \left[\tfrac{n_{\mathrm{bins}}}{2\pi} \left(\arctan2\left(\partial_m \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}, \partial_n \mathbf{v}^{o_{\mathrm{key}}}_{s_{\mathrm{key}},m,n}\right) \bmod 2\pi\right)\right]$$

`// Update the histogram`

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

`// 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}

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

note: $[\cdot]$ denotes the round function and arctan2 denotes the two-argument inverse tangent.

Algorithm 12: Construction of the keypoint descriptor

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

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**

`// Check if the keypoint is distant enough from the image borders`
if $\sqrt{2}\lambda_{descr}\sigma_{key} \le x_{key} \le h - \sqrt{2}\lambda_{descr}\sigma_{key}$ **and** $\sqrt{2}\lambda_{descr}\sigma_{key} \le y_{key} \le w - \sqrt{2}\lambda_{descr}\sigma_{key}$ **then**

`// 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 (Equation (25))`
for $m = \left\lceil \left(x_{key} - \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n_{hist}+1}{n_{hist}}\right)/\delta_o \right\rceil, \dots, \left\lfloor \left(x_{key} + \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n_{hist}+1}{n_{hist}}\right)/\delta_o \right\rfloor$ **do**
for $n = \left\lceil \left(y_{key} - \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n_{hist}+1}{n_{hist}}\right)/\delta_o \right\rceil, \dots, \left\lfloor \left(y_{key} + \sqrt{2}\lambda_{descr}\sigma_{key}\frac{n_{hist}+1}{n_{hist}}\right)/\delta_o \right\rfloor$ **do**

`// Compute normalized coordinates (Equation (24)).`
$\hat{x}_{m,n} = \left((m\delta_{o_{\mathrm{key}}} - x_{\mathrm{key}})\cos\theta_{\mathrm{key}} + (n\delta_{o_{\mathrm{key}}} - y_{\mathrm{key}})\sin\theta_{\mathrm{key}}\right)/\sigma_{\mathrm{key}}$
$\hat{y}_{m,n} = \left(-(m\delta_{o_{\mathrm{key}}} - x_{\mathrm{key}})\sin\theta_{\mathrm{key}} + (n\delta_{o_{\mathrm{key}}} - y_{\mathrm{key}})\cos\theta_{\mathrm{key}}\right)/\sigma_{\mathrm{key}}$

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

`// Update the nearest histograms and the nearest bins (Equation (28)).`
for $(i,j) \in \{1,\dots,n_{hist}\}^2$ *such that* $\left|\hat{x}^i - \hat{x}_{m,n}\right| \le \frac{2\lambda_{descr}}{n_{hist}}$ *and* $\left|\hat{y}^j - \hat{y}_{m,n}\right| \le \frac{2\lambda_{descr}}{n_{hist}}$ **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**
$\mathbf{f}_l \leftarrow \min(\mathbf{f}_l, 0.2\|\mathbf{f}\|)$ `/*normalize and threshold f*/`
compute the l2 norm $\mathbf{f}_l \leftarrow \min(\lfloor 512\mathbf{f}_l/\|\mathbf{f}\|\rfloor, 255)$ `/*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 images. In the absence of extra knowledge on the problem, for instance in the form of geometric constraints, a matching procedure generally consists of two steps: the pairing of similar keypoints from respective images and the selection of those that are reliable. In what follows, we present the matching method described in the original article by D. Lowe [17]. 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,

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

Pairing a keypoint with descriptor $\mathbf{f}^a$ requires then to compute distances to all descriptors in $\mathcal{L}^B$. A pair is considered reliable only if its absolute distance is below a certain threshold $C^{\text{match}}_{\text{absolute}}$. Otherwise it is discarded.

To avoid dependence to an absolute distance, the SIFT method uses the second nearest neighbor to define what constitutes a reliable match. SIFT applies an adaptive thresholding $\|\mathbf{f}^a - \mathbf{f}^{b'}\| C^{\text{match}}_{\text{relative}}$, where $\mathbf{f}_{b'}$ is the second nearest neighbor,

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

This is detailed 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 that case, the distance to the nearest and the second nearest descriptor would be comparable. More sophisticated techniques have been developed to allow robust matching of images with repeated structures [20].

This matching algorithm runs in time $c \cdot N_A \cdot N_B$, where N_A and N_B are the number of keypoints in images $\mathbf{u}^A$ and $\mathbf{u}^B$ respectively, and c is a constant proportional to the time that takes to compare two SIFT features. This is prohibitively slow for images of moderate size, although keypoint matching is highly parallelizable. A better solution is to use more compact descriptors [26] that reduce the cost of distance computation (and thus reduce the value of c). Among the proposed solutions we can find more compact SIFT-like descriptors [2, 12] or binary descriptors [5, 24, 21] which take advantage of the fast computation of the Hamming distance between two binary vectors.

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 summarize all the parameters that can be adjusted in the demo.

Digital scale-space configuration and keypoints detection

Parameter	Default value	Description
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)
n_{oct}	8	number of octaves (limited by the image size)) $\lfloor \log_2(\min(w,h)/\delta_{\text{min}}/12) + 1 \rfloor$
n_{spo}	3	number of scales per octave
C_{DoG}	0.015	threshold over the DoG response (value relative to $n_{\text{spo}} = 3$)
C_{edge}	10	threshold over the ratio of principal curvatures (edgeness).

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

The structure of the digital scale-space 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 the user to change these values. They can be tuned to satisfy specific requirements[4]. For example, increasing the number of scales per octave n_{spo} and the initial interpolation factor δ_{min} increases the number of detections. On the other hand, reducing them results 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} (i.e. $\sigma_{\text{min}} = \sigma_{\text{in}}$). However, depending on the input image sharpness, low scale detections may be the result of aliasing artifacts and should be avoided. Thus, a sound setting of parameter σ_{min} should take into account the image blur level σ_{in} and the possible presence of image aliasing.

The DoG thresholding, controlled by C_{DoG}, was conceived to filter detections due to noise. With that aim in view, C_{DoG} should depend on the input image signal to noise ratio. It is however beyond the scope of this publication to analyze the soundness of such an approach. We will only point out that the reduction of C_{DoG} increases the number of detected keypoints. Recall that the DoG approximates $(2^{1/n_{\text{spo}}} - 1)\sigma^2 \Delta v$, its values depend on the number of scales per octave n_{spo}. The threshold applied in the provided source-code is $\widetilde{C}_{\text{DoG}} = \frac{2^{1/n_{\text{spo}}}-1}{2^{1/3}-1} C_{\text{DoG}}$, with C_{DoG} relative to $n_{\text{spo}} = 3$.

The threshold C_{edge}, applied to discard keypoints laying on edges, has in practice a negligible impact on the algorithm performance. Indeed, many candidate keypoints laying on edges were previously discarded during the extrema refinement.

Computation of the SIFT descriptor

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

The parameter λ_{ori} controls how local the computation of the reference orientation is. Localizing the gradient analysis generally results in an increased number of orientation references. Indeed, the orientation histogram generated from an isotropic structure is almost flat and therefore has

[4] The number of computed octaves is upper limited to ensure that images in the last octave are at least 12×12 pixels.

Parameter	Default 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 $2\lambda_{\text{descr}}\sigma$

Table 4: Parameters of the computation of the SIFT feature vectors.

many local maxima. Another algorithm design parameter, not included in Table 4 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 scene deformation. 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 compact. These architectural parameters govern the trade off between robustness and discrimination.

Matching of SIFT feature vectors

The SIFT algorithm consists of the detection of image keypoints and their description. The demo provides additionally two naive algorithms to match SIFT features. The first one applies an absolute threshold on the distance to the nearest keypoint feature to define if a match is reliable. The second one applies a relative threshold that depends on the distance to the second nearest keypoint feature.

Increasing the absolute threshold $C^{\text{match}}_{\text{absolute}}$ evidently reduces the number of matches. In a relative threshold 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 will be less likely to be omitted. However, this may lead to an increased number of false matches.

Parameter	Default value	Description
$C^{\text{match}}_{\text{absolute}}$	250 to 300	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 SIFT matching algorithm.

Acknowledgements

This work was partially supported by the Centre National d'Etudes Spatiales (CNES, MISS Project), the European Research Council (Advanced Grant Twelve Labours), the Office of Naval Research (Grant N00014-97-1-0839), Direction Générale de l'Armement (DGA), Fondation Mathématique Jacques Hadamard and Agence Nationale de la Recherche (Stereo project).

Image Credits

Crops of stereoscopic cards by T. Enami (1859-1929) were used in figures 3, 6, 8, 9 and 11.

References

[1] L. Alvarez and F. Morales, *Affine morphological multiscale analysis of corners and multiple junctions*, International Journal of Computer Vision, 25 (1997), pp. 95–107. http://dx.doi.org/10.1023/A:1007959616598.

[2] H. Bay, T. Tuytelaars, and L. van Gool, *SURF: Speeded Up Robust Features*, in European Conference on Computer Vision, 2006.

[3] M. Brown, R. Szeliski, and S. Winder, *Multi-image matching using multi-scale oriented patches*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005. http://dx.doi.org/10.1109/CVPR.2005.235.

[4] P J Burt and E H Adelson, *The Laplacian pyramid as a compact image code*, IEEE Transactions on Communications, 31 (1983), pp. 532–540. http://dx.doi.org/10.1109/TCOM.1983.1095851.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, *BRIEF: Binary Robust Independent Elementary Features*, in European Conference on Computer Vision, 2010, pp. 778–792. http://dx.doi.org/10.1007/978-3-642-15561-1_56.

[6] J L Crowley and R M Stern, *Fast computation of the difference of low-pass transform*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1984), pp. 212–222. http://dx.doi.org/10.1109/TPAMI.1984.4767504.

[7] Wolfgang Förstner, *A framework for low level feature extraction*, in European Conference on Computer Vision, 1994, pp. 383–394. http://dx.doi.org/10.1007/BFb0028370.

[8] W. Förstner, T. Dickscheid, and F. Schindler, *Detecting interpretable and accurate scale-invariant keypoints*, in Proceedings of IEEE International Conference on Computer Vision, 2009. http://dx.doi.org/10.1109/ICCV.2009.5459458.

[9] P Getreuer, *A survey of Gaussian convolution algorithms*, Image Processing On Line, 3 (2013), pp. 286–310. http://dx.doi.org/10.5201/ipol.2013.87.

[10] C. Harris and M. Stephens, *A combined corner and edge detector*, in Alvey Vision Conference, vol. 15, 1988, p. 50. http://dx.doi.org/10.5244/C.2.23.

[11] T. Hassner, V. Mayzels, and L. Zelnik-Manor, *On SIFTs and their scales*, in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1522–1528. http://dx.doi.org/10.1109/CVPR.2012.6247842.

[12] Y. Ke and R. Sukthankar, *PCA-SIFT: A more distinctive representation for local image descriptors*, in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2004. http://dx.doi.org/10.1109/CVPR.2004.1315206.

[13] S. Leutenegger, M. Chli, and R.Y. Siegwart, *BRISK: Binary Robust Invariant Scalable Keypoints*, in Proceedings of IEEE International Conference on Computer Vision, 2011. http://dx.doi.org/10.1109/ICCV.2011.6126542.

[14] T. Lindeberg, *Scale-space theory in computer vision*, Springer, 1993. http://dx.doi.org/10.1007/978-1-4757-6465-9.

[15] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman, *SIFT Flow: Dense correspondence across different scenes*, in European Conference on Computer Vision, 2008. http://dx.doi.org/10.1007/978-3-540-88690-7_3.

[16] D. Lowe, *Object recognition from local scale-invariant features*, in IEEE International Conference on Computer Vision, 1999. http://dx.doi.org/10.1109/ICCV.1999.790410.

[17] ——, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[18] E. Mair, G.D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, *Adaptive and generic corner detection based on the accelerated segment test*, in European Conference on Computer Vision, Springer, 2010, pp. 183–196. http://dx.doi.org/10.1007/978-3-642-15552-9_14.

[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, *A comparison of affine region detectors*, International Journal of Computer Vision, 65, pp. 43–72. http://dx.doi.org/10.1007/s11263-005-3848-x.

[20] J. Rabin, J. Delon, and Y. Gousseau, *A statistical approach to the matching of local features*, SIAM Journal on Imaging Science, 2 (2009), pp. 931–958. http://dx.doi.org/10.1137/090751359.

[21] M. Raginsky and S. Lazebnik, *Locality-sensitive binary codes from shift-invariant kernels*, in Advances in Neural Information Processing Systems, vol. 22, 2009, pp. 1509–1517.

[22] I. Rey-Otero and M. Delbracio, *Computing an exact Gaussian scale-space*. Preprint, http://www.ipol.im/pub/pre/117, 2014.

[23] E. Rosten and T. Drummond, *Machine learning for high-speed corner detection*, in European Conference on Computer Vision, 2006. http://dx.doi.org/10.1007/11744023_34.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, *ORB: An efficient alternative to SIFT or SURF*, in Proceedings of IEEE International Conference on Computer Vision, 2011, pp. 2564–2571. http://dx.doi.org/10.1109/ICCV.2011.6126544.

[25] S.M. Smith and J.M. Brady, *SUSAN. A new approach to low level image processing*, International Journal of Computer Vision, 23 (1997), pp. 45–78. http://dx.doi.org/10.1023/A:1007963824710.

[26] T. Tuytelaars and K. Mikolajczyk, *Local invariant feature detectors: A survey*, Foundations and Trends in Computer Graphics and Vision, 3 (2008), pp. 177–280. http://dx.doi.org/10.1561/0600000017.

[27] J. Weickert, S. Ishikawa, and A. Imiya, *Linear scale-space has first been proposed in Japan*, Journal of Mathematical Imaging and Vision, 10 (1999), pp. 237–252. http://dx.doi.org/10.1023/A:1008344623873.

[28] G. Yu and J-M. Morel, *ASIFT: An Algorithm for Fully Affine Invariant Comparison*, Image Processing On Line, 1 (2011). http://dx.doi.org/10.5201/ipol.2011.my-asift.