An implementation of SIFT detector and descriptor

Andrea Vedaldi
University of California at Los Angeles

Contents

1 Introduction

These notes describe an implementation of the Scale-Invariant Transform Feature (SIFT) detector and descriptor [1]. The implementation is designed to produce results compatible to Lowe's version.[1] Designed for the MATLAB environment, it is broken down into several M and MEX files that enable running only portion of the algorithm.

The SIFT *detector* extracts from an image a collection of *frames* or *keypoints*. These are oriented disks attached to blob-alike structures of the image. As the image translates, rotates and scales, the frames track these blobs and thus the deformation. By *canonization*, i.e. by mapping the frames to a reference (a canonical disk), the effect of such deformation on the feature appearance is removed.

The SIFT *descriptor* is a coarse description of the edge found in the frame. Due to canonization, descriptors are invariant to translations, rotations and scalings and are designed to be robust to residual small distortions.

The SIFT detector and descriptor are discussed in depth in [1]. Here we only describe the interface to our implementation and, in the Appendix, some technical details.

2 User reference: the `sift` function

The SIFT detector and the SIFT descriptor are invoked by means of the function `sift`, which provides a unified interface to both.

Example 1 (Invocation). The following lines run the SIFT detector and descriptor on the image `data/test.jpg`.

[1]See `http://www.cs.ubc.ca/~lowe/keypoints/`

```
I = imread('data/test.png') ;
I = double(rgb2gray(I)/256) ;
[frames,descriptors] = sift(I, 'Verbosity', 1) ;
```

The pair option-value `'Verbosity',1` causes the function to print a detailed progress report.

The `sift` function returns a $4 \times K$ matrix `frames` containing the SIFT frames and a $128 \times K$ matrix `descriptors` containing their descriptors. Each frame is characterized by four numbers which are in order (x_1, x_2) for the center of the frame, σ for its scale and θ for its orientation. The coordinates (x_1, x_2) are relative to the upper-left corner of the image, which is assigned coordinates $(0, 0)$, and may be fractional numbers (sub-pixel precision). The scale σ is the smoothing level at which the frame has been detected. This number can also be interpreted as size of the frame, which is usually visualized as a disk of radius 6σ. Each descriptor is a vector describing coarsely the appearance of the image patch corresponding to the frame (further details are discussed in Appendix A.3). Typically this vector has dimension 128, but this number can be changed by the user as described later.

Once frames and descriptors of two images I_1 and I_2 have been computed, `siftmatch` can be used to estimate the pairs of matching features. This function uses Lowe's method to discard ambiguous matches [1]. The result is a $2 \times M$ matrix, each column of which is a pair (k_1, k_2) of indices of corresponding SIFT frames.

Example 2 (Matching). Let us assume that the images `I1` and `I2` have been loaded and processed as in the previous example. The code

```
matches = siftmatch(descriptors1, descriptors2) ;
```

stores in `matches` the matching pairs, one per column.

The package provides some ancillary functions; you can

- use `plotsiftframe` to plot SIFT frames;
- use `plotsiftdescriptor` to plot SIFT descriptors;
- use `plotmatches` to plot feature matches;
- use `siftread` to read files produced by Lowe's implementation.

Example 3 (Visualization). Let `I1`, `I2` and `matches` be as in the previous example. To visualize the matches issue

```
plotsiftmatches(I1,I2,frames1,frames2,matches)
```

The `sift` function has many parameters. The default values have been chosen to emulate Lowe's original implementation. Although our code does not result in frames and descriptors that are 100% equivalent, in general they are quite similar.

2.1 Scale space parameters

The SIFT detector and descriptor are constructed from the *Gaussian scale space* of the source image $I(x)$. The Gaussian scale space is the function

$$G(x; \sigma) \triangleq (g_\sigma * I)(x)$$

where g_σ is an isotropic Gaussian kernel of variance $\sigma^2 I$, x is the spatial coordinate and σ is the scale coordinate. The algorithm make use of another scale space too, called *difference of Gaussian (DOG)*, which is, coarsely speaking, the scale derivative of the Gaussian scale space.

Since the scale space $G(x; \sigma)$ represents the same information (the image $I(x)$) at different levels of scale, it is sampled in a particular way to reduce redundancy. The domain of the variable σ is discretized in logarithmic steps arranged in O octaves. Each octave is further subdivided in S sub-levels. The distinction between octave and sub-level is important because at each successive octave the data is spatially downsampled by half. Octaves and sub-levels are identified by a discrete *octave index* o and *sub-level index* s respectively. The octave index o and the sub-level index s are mapped to the corresponding scale σ by the formula

$$\sigma(o,s) = \sigma_0 2^{o+s/S}, \quad o \in o_{\min} + [0,...,O-1], \quad s \in [0,...,S-1] \tag{1}$$

where σ_0 is the base scale level.

The `sift` function accepts the following parameters describing the Gaussian scale space being used:

- `NumOctaves`. This is the number of octaves O in (1).
- `FirstOctave`. Index of the first octave $o_{\min}$: the octave index o varies in $o_{\min},...,o_{\min}+O-1$. It is usually either 0 or -1. Setting $o_{\min}$ to -1 has the effect of doubling the image before computing the Gaussian scale space.
- `NumLevels`. This is the number of sub-levels S in (1).
- `Sigma0`. Base smoothing: This is the parameter σ_0 in (1).
- `SigmaN`. Nominal pre-smoothing: This is the nominal smoothing level of the input image. The algorithm assumes that the input image is actually $(g_{\sigma_n} * I)(x)$ as opposed to $I(x)$ and adjusts the computations according. Usually σ_n is assumed to be half pixel (0.5).

2.2 Detector parameters

The SIFT frames (x, σ) are points of local extremum of the DOG scale space. The selection of such points is controlled by the following parameters:

- `Threshold`. Local extrema threshold. Local extrema whose value $|G(x,;\sigma)|$ is below this number are rejected.
- `EdgeThreshold`. Local extrema localization threshold. If the local extremum is on a valley, the algorithm discards it as it is too unstable. Extrema are associated with a score proportional to their sharpness and rejected if the score is below this threshold.
- `RemoveBoundaryPoints`. Boundary points removal. If this parameter is set to 1 (true), frames which are too close to the boundary of the image are rejected.

2.3 Descriptor parameters

The SIFT descriptor is a weighted and interpolated histogram of the gradient orientations and locations in a patch surrounding the keypoint. The descriptor has the following parameters:

- `Magnif`. Magnification factor m. Each spatial bin of the histogram has support of size $m\sigma$, where σ is the scale of the frame.
- `NumSpatialBins`. Number of spatial bins. Together with the next parameter, this number defines the extension and dimension of the descriptor. The dimension of the descriptor (the total number of bins) is equal to $\texttt{NumSpatialBins}^2 \times \texttt{NumOrientBins}$ and its extension (the patch where the gradient statistic is collected) has radius $\texttt{NumSpatialBins} \times m\sigma/2$.
- `NumOrientBins`. Number of orientation bins.

Symbol	Description	In the code
$o \in [o_{\min}, o_{\min} + O - 1]$	Octave index and range	`o, O, omin`
$s \in [s_{\min}, s_{\max}]$	Scale index and range	`s, smin, smax`
$\sigma(o, s) = \sigma_0 2^{o+s/S}$	Scale coordinate formula	
σ_0	Base scale offset	`sigma0`
M_0, N_0	Base spatial resolution (octave $o = 0$)	
$N_o = \lfloor \frac{N_0}{2^o} \rfloor$, $M_o = \lfloor \frac{M_0}{2^o} \rfloor$	Octave lattice size formulas	
$x_o \in [0, ..., N_o] \times [0, ..., M_o]$	Spatial indexes and rages	
$x = 2^o x_o$	Spatial coordinate formula	
$F(\cdot, \sigma(o, \cdot))$	Octave data	`octave`
$G(x, \sigma)$	Gaussian scale space	`gss`
$D(x, \sigma)$	DOG scale space	`dogss`

Figure 1: *Scale space parameters.* The SIFT descriptors uses two scale spaces: a Gaussian scale space and a Difference of Gaussian scale space. Both are described by these parameters.

2.4 Direct access to SIFT components

The SIFT code is decomposed in several M and MEX files, each implementing a portion of the algorithm. These programs can be run on their own or replaced. Appendix A contains information useful to do this.

Example 4 (Computing the SIFT descriptor directly). Sometimes it is useful to run the descriptor code alone. This can be done by calling the function `siftdescriptor` (which is actually a MEX file.) See the function help for further details.

References

[1] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.

A Internals

A.1 Scale spaces

Here a *scale space* is a function $F(x, \sigma) \in \mathbb{R}$ of a spatial coordinate $x \in \mathbb{R}^2$ and a scale coordinate $\sigma \in \mathbb{R}_+$. Since a scale space $F(\cdot, \sigma)$ typically represents the same information at various scales $\sigma \in \mathbb{R}$, its domain is sampled in a particular way in order to reduce the redundancy.

The scale coordinate σ is discretized in logarithmic steps according to

$$\sigma(s, o) = \sigma_0 2^{o+s/S}, \qquad o \in \mathbb{Z}, \quad s = 0, ..., S - 1$$

where o is the *octave index*, s is the *scale index*, $S \in \mathbb{N}$ is the *scale resolution* and $\sigma_0 \in \mathbb{R}_+$ is the *base scale offset*. Note that it is possible to have octaves of negative index.

The spatial coordinate x is sampled on a lattice with a resolution which is a function of the octave. We denote x_o the spatial index for octave o; this index is mapped to the coordinate x by

$$x = 2^o x_o, \qquad o \in \mathbb{Z}, \quad x_o \in [0, ..., N_o - 1] \times [0, ..., M_o - 1].$$

where (N_o, M_o) is the spatial resolution of octave o. If (M_0, N_0) is the the resolution of the base octave $o = 0$, the resolution of the other octaves is obtained as

$$N_o = \lfloor \frac{N_0}{2^o} \rfloor, \qquad M_o = \lfloor \frac{M_0}{2^o} \rfloor.$$

It will be useful to store some scale levels twice, across different octaves. We do this by allowing the parameter s to be negative or greater than S. Formally, we denote the range of s as $[s_{\min}, s_{\max}]$. We also denote the range of the octave index o as $[o_{\min}, o_{\min} + O - 1]$, where $O \in \mathbb{N}$ is the total number of octaves. See Figure 1 for a summary of these symbols.

The SIFT detector makes use of the two scale spaces described next.

Gaussian Scale Space. The *Gaussian scale space* of an image $I(x)$ is the function

$$G(x, \sigma) \triangleq (g_\sigma * I)(x)$$

where the scale $\sigma = \sigma_0 2^{o+s/S}$ is sampled as explained in the previous section. This scale space is computed by the function `gaussianss`. In practice, it is assumed that the image passed to the function `gaussianss` is already pre-smoothed at a nominal level σ_n, so that $G(x, \sigma) = (g_{\sqrt{\sigma^2 - \sigma_n^2}} * I)(x)$. As suggested in [1], the pyramid is computed incrementally from the bottom by successive convolutions with small kernels.

Difference of Gaussians scale space. The *difference of Gaussians (DOG) scale space* is the scale "derivative" of the Gaussian scale space $G(x, \sigma)$ along the scale coordinate σ. It is given by

$$D(x, \sigma(s, o)) \triangleq G(x, \sigma(s + 1, o)) - G(x, \sigma(s, o)).$$

It is obtained from the Gaussian scale space by the `diffss` function.

Remark 1 (Lowe's parameters). Lowe's implementation uses the following parameters:

$$\sigma_n = 0.5, \quad \sigma_0 = 1.6 \cdot 2^{1/S}, \quad o_{\min} = -1, \quad S = 3$$

In order to compute the octave $o = -1$, the image is doubled by bilinear interpolation (for the enlarged image $\sigma_n = 1$). In order to detect extrema at all scales, the Difference of Gaussian scale space has $s \in [s_{\min}, s_{\max}] = [-1, S]$. Since the Difference of Gaussian scale space is obtained by differentiating the Gaussian scale space, the latter has $s \in [s_{\min}, s_{\max}] = [-1, S + 1]$. The parameter O is set to cover all octaves (i.e. as big as possible.)

A.2 The detector

The SIFT frames (or "keypoints") are a selection of (sub-pixel interpolated) points (x, σ) of local extremum of the DOG scale-space $D(x, \sigma)$, together with an orientation θ derived from the spatial derivative of the Gaussian scale-space $G(x, \sigma)$. For what concerns the detector (and being in general different for the descriptor), the "support" of a keypoint (x, σ) is a Gaussian window $H(x)$ of deviation $\sigma_w = 1.5\sigma$. In practice, the window is truncated at $|x| \leq 4\sigma_w$.

The Gaussian and DOG scale spaces are derived as in Section A.1. In this Section, the parameters S, O, $s_{\min}$, $s_{\max}$, $o_{\min}$, σ_0 refer to the DOG scale space. The Gaussian scale space has exactly the same parameters of the DOG scale space except for $s_{\max}^{\text{DOG}}$ which is equal to $s_{\max} - 1$.

The extraction of the keypoints is carried one octave per time and articulated in the following steps:

- **Detection.** Keypoints are detected as points of local extremum of $D(x, \sigma)$ (Section A.1). In the implementation the function `siftlocalmax` extracts such extrema by looking at $9 \times 9 \times 9$ neighborhoods of samples.

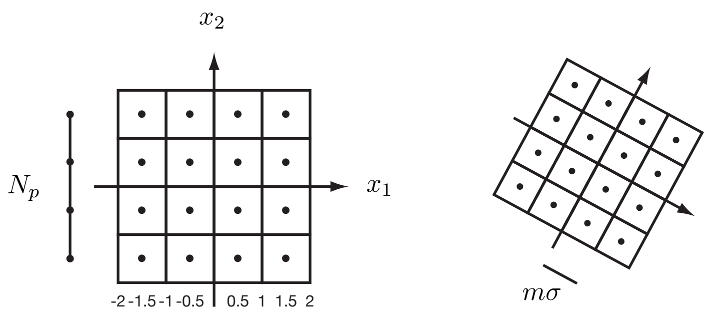

Figure 2: *SIFT descriptor layout.* The actual size of a spatial bin is $m\sigma$ where σ is the scale of the keypoint and $m = 3.0$ is a nominal factor.

As the octave is represented by a 3D array, the function `siftlocalmax` returns indexes k (in MATLAB convetion) that are to be mapped to scale space indexes (x_1, x_2, s) by

$$k - 1 = x_2 + x_1 M_o + (s - s_{\min}) M_o N_o.$$

Alternatively, one can use `ind2sub` to map the index k to a subscript (i, j, l) and then use

$$x_1 = j - 1, \qquad x_2 = i - 1, \qquad s = l - 1 + s_{\min}.$$

Because of the way such maxima are detected, one has always $1 \leq x_2 \leq M_o - 2$, $1 \leq x_1 \leq N_o - 2$ and $s_{\min} + 1 \leq s \leq s_{\max} - 1$.

Since we are interested both in local maxima and minima, the process is repeated for $-G(x, \sigma)$. (If only positive maxima and negative minima are of interest, another option is to take the local maxima of $|G(x, \sigma)|$ directly, which is quicker.)

- **Sub-pixel refinement.** After being extracted by `siftlocalmax`, the index (x_1, x_2, s) is fitted to the local extremum by quadratic interpolation. At the same time, a threshold on the "intensity" $D(x, \sigma)$ and a test on the "peakedness" of the extremum is applied in order to reject weak points or points on edges. These operations are performed by the `siftrefinemx` function.

  The edge rejection step is explained in detail in the paper [1]. The sub-pixel refinement is an instance of Newton's algorithm.

- **Orientation.** The orientation θ of a keypoint (x, σ) is obtained as the predominant orientation of the gradient in a window around the keypoint. The predominant orientation is obtained as the (quadratically interpolated) maximum of the histogram of the gradient orientations $\angle \nabla G(x_1, x_2, \sigma)$ within a window around the keypoint. The histogram is weighted both by the magnitude of the gradient $|\nabla G(x_1, x_2, \sigma)|$ and a Gaussian window centered on the keypoint and of deviation 1.5σ (the Gaussian window defines the region of interest as well). After collecting the data in the bins and before computing the maximum, the histogram is smoothed by a moving average filter.

  In addition to the global maximum, each local maximum with a value above 0.8% of the maxium is retained as well. Thus for each location and scale multiple SIFT frames might be generated.

  These computations are carried by the function `siftormx`.

A.3 The descriptor

The SIFT descriptor of a keypoint (x, σ) is a local statistic of the orientations of the gradient of the Gaussian scale space $G(\cdot, \sigma)$.

Histogram layout. The SIFT descriptor (Figure 2) is an histogram of the gradient orientations augmented and 2-D locations in the support of the SIFT frame. Formally, the domain of the histogram are the tuples $(x, \theta) \in \mathbb{R}^2 \times \mathbb{R}/\mathbb{Z}$. The bins form a three dimensional lattice with $N_p = 4$ bins for each spatial direction and $N_o = 8$ bins for the orientation for a total of $N_p^2 N_o = 128$ components (these numbers can be changed by setting the appropriate parameters). Each spatial bin is square with unitary edge. The window $H(x)$ is Gaussian with deviation equal to half the extension of the spatial bin range, that is $N_p/2$.

Keypoint normalization. In order to achieve invariance, the histogram layout is projected on the image domain according to the frame of reference of the keypoint. The spatial dimensions are multiplied by $m\sigma$ where σ is the scale of the keypoint and m is a nominal factor (equal to 3.0 by default). The layout is also rotated so that the axis x_1 is aligned to the direction θ of the keypoint.

Weighting. The histogram is weighted by the gradient modulus and a Gaussian windowed and tri-linearly interpolated. More in detail, each sample $(x_1, x_2, \angle\nabla G(x, \sigma))$ is

- weighted by $|\nabla G(x, \sigma)|$;
- weighted by the gaussian window $H(x)$;
- projected on the centers of the eight surrounding bins;
- summed to each of this bins proportionally to its distance from the respective center.

Remark 2. (Lowe's impelmentation) In order to achieve full compatibility with Lowe's original implementation, one needs to pay attention to many little details as the memory layout of the descriptor and the convention for the gradient orientations. These are detailed in the source code.