

Fingerprint matching using sift features

Chapter 2 -Literature Review



|  |  |
| --- | --- |
| ID | FULL NAME |
|  |  |
|  |  |

Contents

[Fingerprint Algorithms 2](#_Toc105756608)

[Scale-Invariant Feature Transformation 2](#_Toc105756609)

[The Gaussian Scale-Space 2](#_Toc105756610)

[Gaussian Blurring 2](#_Toc105756611)

[Key Point Definition 3](#_Toc105756612)

[Difference of Gaussians 3](#_Toc105756613)

[Extraction of Candidate Points (Keypoint Localization) 3](#_Toc105756614)

[Filtering Unstable Key Points 3](#_Toc105756615)

[Orientation Assignment 3](#_Toc105756616)

[Key point Descriptor 4](#_Toc105756617)

[Matching 5](#_Toc105756618)

[SIFT on Fingerprint Images 5](#_Toc105756619)

[Preprocessing (Obtaining scale space) 5](#_Toc105756620)

[Local Extrema & Descriptor Extraction 6](#_Toc105756621)

[Point Wise Matching 8](#_Toc105756622)

[Algorithm we will use against SIFT 9](#_Toc105756623)

[What is It 9](#_Toc105756624)

[Characteristics 9](#_Toc105756625)

[How it works 9](#_Toc105756626)

[Performance 9](#_Toc105756627)

# Fingerprint Algorithms

## Scale-Invariant Feature Transformation

Scale-Invariant Feature Transformation (SIFT) is a computer vision algorithm to describe, detect and match local features in digital images. It locates certain key points and then furnishes them with quantitative information or descriptors that can be used for object recognition. These descriptors are invariant against various transformations such as image translation, rotation and scaling which might make images look different although they represent the same object.( [SIFT - Scale-Invariant Feature Transform (weitz.de)](http://weitz.de/sift/)) SIFT descriptors have also proved to be robust to a wide family of image transformations including viewport changes, noise, blur, scene deformation and contrast changes while remaining discriminative enough for matching purposes. The Scale-Invariant Feature Transform (SIFT) algorithm consists of two successive and independent operations. These are

1. The detection of interesting points (key points)
2. The extraction of a descriptor associated to each of them.

Since these descriptors are robust, they are usually used for matching pairs of images. ( [\*article.pdf](file:///C:\Users\Andy\Downloads\Documents\article.pdf)) . Video stabilization is another popular application of the SIFT method, however, the scope of this research will be limited to image recognition, **more specifically finger print image recognition**. **(KEYPOINT SECTION STARTED FROM HERE)** SIFT detects a series of key points from multiscale image representation. This multiscale representation consists of a set of increasingly blurred images. Each key point is a blob-like structure whose center position and characteristic scale are accurately located. The dominant orientation over a region which surrounds one of these key points. For each key point, the size, center and orientation are normalized. Due to this normalization the key points also remain invariant to any translation, scale change or rotation.

### The Gaussian Scale-Space

In order to attain this scale invariance, SIFT is built on a gaussian scale space. A Gaussian Scale-Space is a multiscale image representation simulating the family of all possible zoom-outs through increasingly blurred versions of the input image ( [\*article.pdf](file:///C:\Users\Andy\Downloads\Documents\article.pdf)). This blurring process simulates the loss of detail produced when a scene is photographed from farther and farther. 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. (**YOU CAN EXPLAIN MORE ABOUT WHAT GAUSSIAN SCALE SPACE IS).**

A scale space is constructed by applying a variable gaussian operator on an input image. **The Difference of Gaussian (DOG)** images are obtained by subtracting subsequent scales in each octave. Octaves are the set of Gaussian-Smoothed images and image.

### Gaussian Blurring

In the realm of computing, the gaussian scale space is defined by a series of blurred images using the Gaussian Blur or Gaussian smoothing technique by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise or image graininess. However, in computer-vision-based algorithms. It is used as a preprocessing technique in order to enhance images at different scales( [Gaussian blur - Wikipedia](https://en.wikipedia.org/wiki/Gaussian_blur)). In other areas of computer vision, the Gaussian blur is also used as a way to detect edges. Since most edge detection algorithms are sensitive to noise, the gaussian blur serves as a way to reduce the noise in order to make edge detection more accurate. This is a technique the SIFT algorithm uses in order to properly prepare images for feature point detection. In our case, it will be used to prepare fingerprint images for the detection of where the fingerprint arc, whorl or loop begins.

### Key Point Definition

In SIFT, key points are defined as the **3D extrema** of the normalized scale-space. The extrema are detected by observing each image point in the Difference of Gaussian (DoG) space. The Local Extrema are detected by observing each image point in the Difference of Gaussian Space. A point is decided as a local minimum or maximum when its value is smaller or larger than all its surrounding neighbor points by a certain amount. If an extremum is decided as unstable or is placed on an edge. It is removed because it can not be reliably detected again with small variation of the viewport or lighting changes.

### Difference of Gaussians

Difference of Gaussians (DoG) here refers to a feature enhancement technique that involves the subtraction of one Gaussian Blurred version of an original image from another less blurred version of the original ([Difference of Gaussians - Wikipedia](https://en.wikipedia.org/wiki/Difference_of_Gaussians)). This technique ensures that the spatial information that lie between the (range of frequencies) are preserved between the blurred images, these include visibility of edges and any other key points present in the digital image. **(TLDR it is used to enhance edges to extract key points)**

### Extraction of Candidate Points (Keypoint Localization)

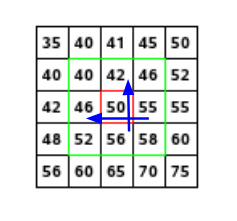
Continuous 3D extrema of the digital DoG are calculated in two steps. Firstly, the 3D discrete extrema are first extracted from each octave with pixel precision. Then their location is refined through interpolation of the digital DoG by using a quadratic model. The resulting image is then compared to its neighbors to detect the 3D discrete maxima and minima. These comparisons are possible due to the auxiliary images in the DoG. Although this process produces candidate points, we can work with, it is prone to noise and as such produces unstable detections and the key points chosen may be flawed since it is constrained to the sampling grid.

### Filtering Unstable Key Points

Noisy images produce erroneous candidate key points thereby making them unstable and unlinked to any particular structure in the image. SIFT attempts to eliminate these false detections by discard those candidate key points found outside the DoG threshold **(STATE WAY TO CALCULATE THRESHOLD).** Another unstable key-points are those on the edges of the image. These candidate key points are difficult to precisely locate due to the fact that an edge is invariant to translations along its principal axis. Such detections do not help define covariant key points and are also discarded.

### Orientation Assignment

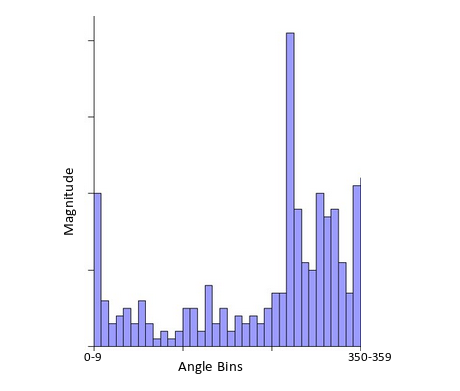
At this point, a set of stable keypoints have been generated and as such, an orientation can be assigned to each of these keypoints to make them invariant to rotation [SIFT | How To Use SIFT For Image Matching In Python (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=SIFT%20helps%20locate%20the%20local,detection%2C%20scene%20detection%2C%20etc.). This is done by computing the magnitude and orientation for each keypoint and then followed by a histogram construction with which we can determine the peak orientation for that particular keypoint. An example is shown below



**LABEL FIGURE**

Consider the matrix above which is a matrix of pixels. To compute the orientation and magnitude for the pixel in red, the gradients in both the and directions are calculated as follows

The magnitude represents the intensity of the pixel while the orientation represents the direction of the pixel. From this a histogram is created by plotting the magnitude and orientation value for all the pixels. An example is shown below;



(**LABEL FIGURE)**

At some point the histogram peaks and from this the orientation of the keypoint is determined making it invariant to rotations.

### Key point Descriptor

In SIFT, descriptors refer to the use of neighboring pixels, their orientations and magnitudes to generate a unique key point.( [SIFT | How To Use SIFT For Image Matching In Python (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=SIFT%20helps%20locate%20the%20local,detection%2C%20scene%20detection%2C%20etc.)). These descriptors fall into two categories, that is

1. Those based on properties of the image that are already rotation-invariant
2. Descriptors based on a normalization with respect to the reference orientation

For the scope of this project, we will use descriptors based on category **(i) (MAYBE EXPLAIN MORE?)**

Keypoints between two images are matched by identifying their nearest neighbors. In the event that the keypoints are too close to each other due to image noise, the ratio of the closest distance to second closest distance is taken. The standard ratio for this distance is 0.8 and if they are greater than this, the points are rejected. This ensures that 90% of false matches are eliminated while only discarding 5% of correct matches. [Introduction to SIFT( Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium](https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40)

### Matching

Following the filtering we can now use these descriptors to identify images or to compare images, the following are reasons why using these descriptors work

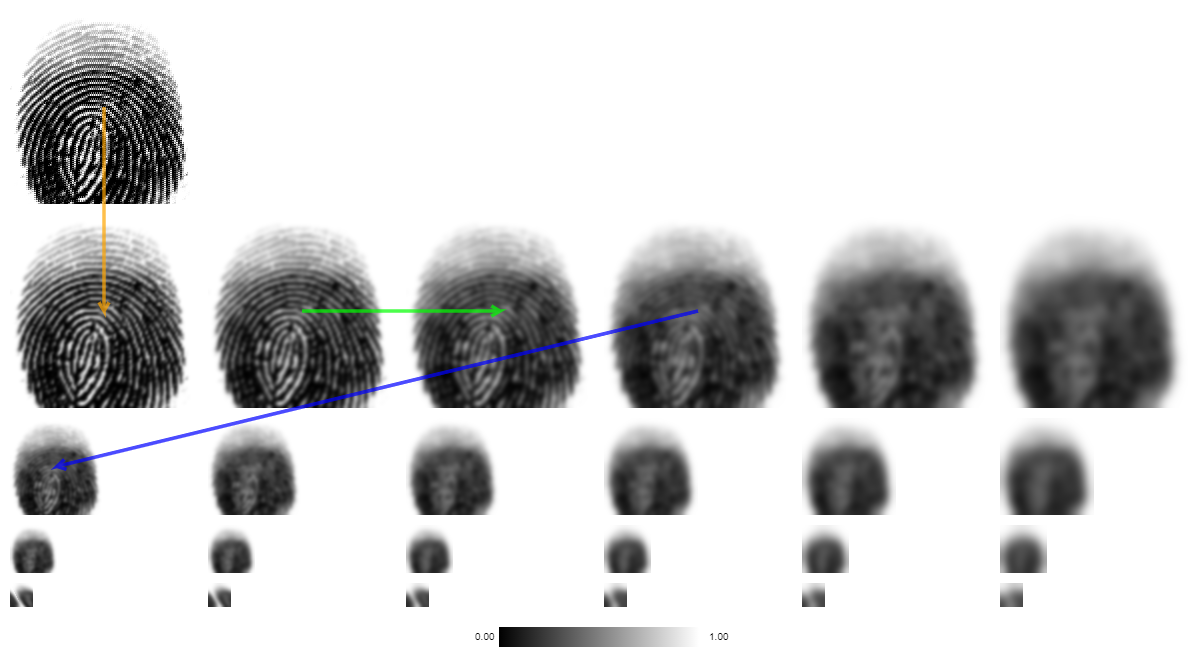
* Key points are extracted at different scales and blur levels and all subsequent computations are performed within the scale space framework. This will make the descriptors invariant to image scaling and small changes in perspective.
* Computation relative to a reference orientation is supposed to make the descriptors robust against rotation.
* Likewise, the descriptor information is stored relative to the key point position and thus invariant against translations.
* Many potential key points are discarded if they are deemed unstable or hard to locate precisely. The remaining key points should thus be relatively immune to image noise.
* The histograms are normalized at the end which means the descriptors will not store the magnitudes of the gradients, only their relations to each other. This should make the descriptors invariant against global, uniform illumination changes.
* The histogram values are also thresholded to reduce the influence of large gradients. This will make the information partly immune to local, non-uniform changes in illumination.
* [SIFT - Scale-Invariant Feature Transform (weitz.de)](http://weitz.de/sift/index.html?size=large)

## SIFT on Fingerprint Images

Since SIFT points are limited by the condition of the local minima or maxima in a given scale space, a large number of feature points are detected. The number of SIFT points are determined by a set of parameters including the number of octaves. Typical fingerprints contain up to a thousand sift points [ParkFingerSIFT\_SPIE2008.pdf](file:///C:\Users\Andy\OneDrive\Desktop\FingerPrint%20Matching%20Project\Resources\ParkFingerSIFT_SPIE2008.pdf). SIFT on fingerprint images can be summarized into four (4) steps. Summarized in the flow chart below

### Preprocessing (Obtaining scale space)

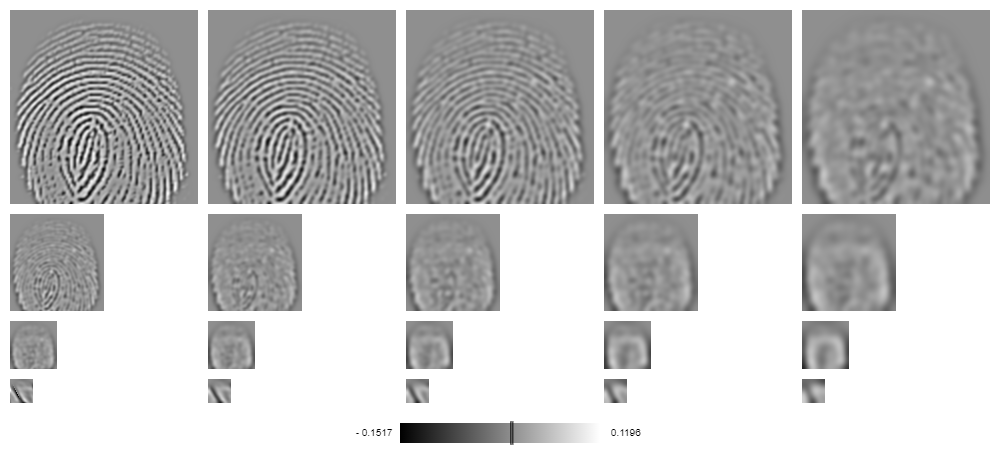
The first phase which involves obtaining the scale space of the fingerprint image which is done by blurring the images using a Gaussian blurring method to simulate the different zoom levels of the image indicated by the orange arrow (**Insert Orange arrow here**) further blurring is done indicated as its left neighbor or by the green arrow (**Insert green arrow here)**. To further make the scale space robust, we can (**Downsample**) the images (**to simulate something indicated by the blue arrow).** Each row is known as an **octave**. Image representation of generating scale space shown below.



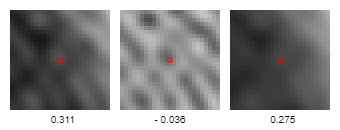
(THE VARIOUS PHASES)

### Local Extrema & Descriptor Extraction

To find the local Extrema in this phase, we first compute the Difference of Gaussian (DoG). This can be done by gray-scaling each image in the octave and calculating the difference between each pixel in the adjacent image

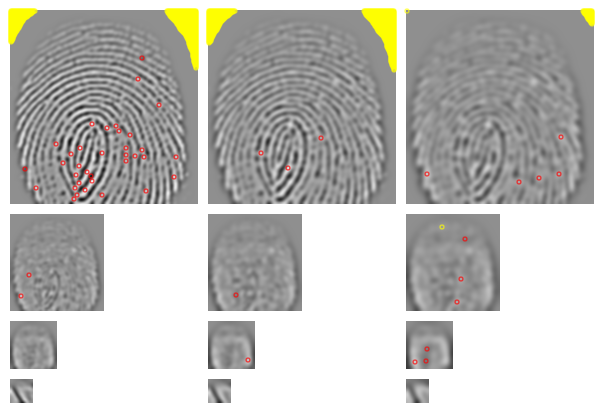


(GRAY SCALING)



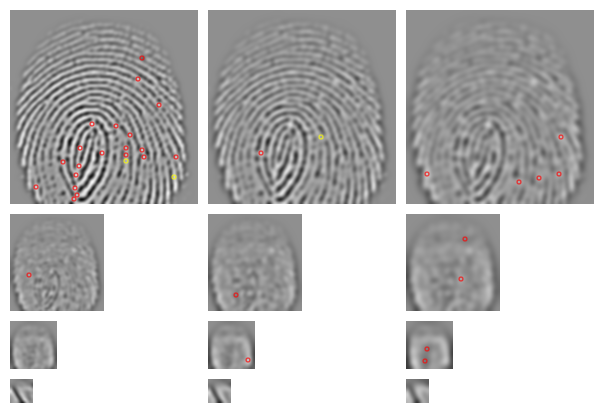
(EXAMPLE OF DIFFERENCE of Gaussian)

The extrema then become a pixel whose gray value is larger than all of its neighboring pixels as shown below.



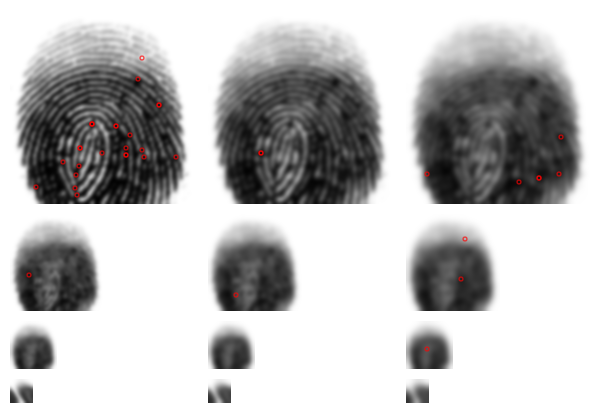
(EXTREMA OBTAINED AFTER USING DoG)

Points marked in yellow are indeed extrema, but their absolute values are so low that we discard them. These extrema usually exist as a result of image noise. Now that we have these extrema, we filter them using the quadratic **Taylor expansion** of scale space-function. This is an iterative process that refines the location of a keypoint. Following the first filtering, we now identify the keypoints that lie on the edges, these points are invariant to translations parallel to edge direction hence we discard them. The remaining key points are shown below



(REFINED KEYPOINTS)

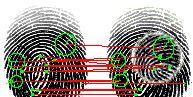
From here, a reference orientation (**Add links to link back to certain things)** is calculated making the keypoints invariant to rotation refer to (**Orientation Assignment**) resulting in descriptors. Descriptors that do not have enough pixels to compute a reference orientation are discarded. Descriptors without a dominating orientation are also discarded resulting in the following image



(AFTER DISCARDING NON-ORIENTED POINTS)

### Point Wise Matching

Now that we have a potentially larger set of descriptors, we can compare it to the descriptors of another image even if they are depicted with different illumination, slightly distorted or with a different perspective. For reasons why this works, refer to (MATCHING). The same process is repeated for the image we are comparing it to, the image below depicts the image sample we used in this example against a distorted version of that same image



Matching Using Descriptor Features

## Algorithm we will use against SIFT

### What is It

### Characteristics

### How it works

### Performance