

Fingerprint matching using sift features

Chapter 2 -Literature Review



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

Contents

[Fingerprint Matching Algorithms 1](#_Toc106202030)

[Scale-Invariant Feature Transformation 1](#_Toc106202031)

[Descriptors 1](#_Toc106202032)

[Keypoints 1](#_Toc106202033)

[Procedures involved in obtaining keypoints and descriptors 2](#_Toc106202034)

[The Gaussian Scale-Space Construction 2](#_Toc106202035)

[Difference of Gaussians 2](#_Toc106202036)

[3D extrema 2](#_Toc106202037)

[Calculation of 3D extrema 2](#_Toc106202038)

[Filtering Unstable Candidate Keypoints 3](#_Toc106202039)

[Orientation Assignment 3](#_Toc106202040)

[Keypoint Descriptor 4](#_Toc106202041)

[Matching of Descriptors 4](#_Toc106202042)

[Use of SIFT on Fingerprint Images 5](#_Toc106202043)

[Preprocessing (Obtaining scale space) 5](#_Toc106202044)

[Local Extrema & Descriptor Extraction 6](#_Toc106202045)

[Point Wise Matching 8](#_Toc106202046)

[Algorithm we will use against SIFT 9](#_Toc106202047)

[What is It 9](#_Toc106202048)

[Characteristics 9](#_Toc106202049)

[How it works 9](#_Toc106202050)

[Performance 9](#_Toc106202051)

# Fingerprint Matching Algorithms

A fingerprint matching algorithm is one that revolves around comparing previously stored templates of fingerprints against candidate fingerprints usually for the purpose of authentication[A04\_0201\_2.PDF](file:///C:\Users\Andy\Downloads\Documents\A04_0201_2.PDF). The algorithm allows for a person to be identified or verified

Two types of algorithms are prevalent; these are:

1. Scale-Invariant Feature Transformation, and
2. ………

## 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 keypoints and then furnishes them with **quantitative information** or **descriptors** that can be used for object recognition. [Scale-Invariant Feature Transform - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/computer-science/scale-invariant-feature-transform#:~:text=Scale%2DInvariant%20Feature%20Transform%20(SIFT)%E2%80%94SIFT%20is%20an,Keypoints%20Detection%2C%20and%20Feature%20Description.)

The Scale-Invariant Feature Transform (SIFT) algorithm consists of two successive and independent operations. These are

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

SIFT is useful for fingerprint matching because it produces unique descriptors on each fingerprint making it a convenient technique to match two fingerprint images. Due to the nature of these descriptors, comparison between samples can be undertaken (regardless of how distorted the image being used for comparison is) because a match is still generated.

### Descriptors

**Descriptors are quantitative information about a keypoint.** 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. Since these descriptors are robust, they are usually used for matching pairs of images. ( [\*article.pdf](file:///C:\Users\Andy\Downloads\Documents\article.pdf)) .

### Keypoints

**Keypoints are interesting points whose center position and characteristic scale are accurately located**. For each keypoint, the size, center and orientation are normalized. Due to this normalization the keypoints remain invariant to any translation, scale change or rotation.

SIFT detects a series of keypoints from multiscale image representation. This multiscale representation consists of a set of increasingly blurred images. From these keypoints, descriptors are generated.

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**.

## Procedures involved in obtaining keypoints and descriptors

To obtain these keypoints and descriptors, various steps and frameworks are constructed and used.

### The Gaussian Scale-Space Construction

Keypoints in SIFT are invariant to scale-changes. 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. This scale space is constructed by applying a variable gaussian operator on an input image. This is also known as **Gaussian Blurring**.

**Gaussian blurring** is a technique used by the SIFT algorithm to properly prepare images for scale space construction. This technique is a widely used effectively 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.

The Gaussian blur technique will be used to prepare fingerprint images for the detection of where the fingerprint arc, whorl or loop begins.

### Difference of Gaussians

Another refinement technique used is the Difference of Gaussians **(DoG)**. This 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 results in a set of Gaussian-Smoothed images known as **octaves**

The Difference of Gaussians technique ensures that the spatial information that lie between the range of frequencies are preserved between the two blurred images, these include visibility of edges and any other keypoints present in the digital image.

### 3D extrema

In SIFT, ***candidate keypoints*** 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)**. 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.

### Calculation of 3D extrema

Continuous 3D extrema of the digital DoG are calculated in two steps. Firstly, the 3D discrete extrema are extracted from each octave with pixel precision. Next, their location is refined through interpolation of the digital DoG by using a quadratic model such as the **Taylor expansion**.

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 keypoints** that can be used, it is prone to noise and as such produces unstable detections. The Candidate keypoints chosen may therefore be flawed since they are constrained to the sampling grid.

### Filtering Unstable Candidate Keypoints

Noisy images produce erroneous candidate keypoints thereby making them unstable and unlinked to any particular structure in the image.

SIFT attempts to eliminate these false detections by discarding those candidate keypoints found outside the DoG threshold on distance ratiodefined as *(distance to nearest candidate keypoint neighbor / distance to second nearest candidate keypoint neighbor)***.**

Unstable key-points are those on the edges of the image. These candidate keypoints 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 keypoints and are also discarded.

### Orientation Assignment

After a set of stable candidate keypoints have been generated, an orientation is 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 constructing a histogram to determine the peak orientation for a particular keypoint.

An example is shown below in ***Figure 1***

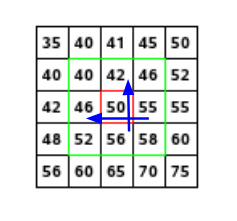


Figure depicting pixel matrix

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 to obtain the peak orientation. An example is shown below in ***Figure 2****;*

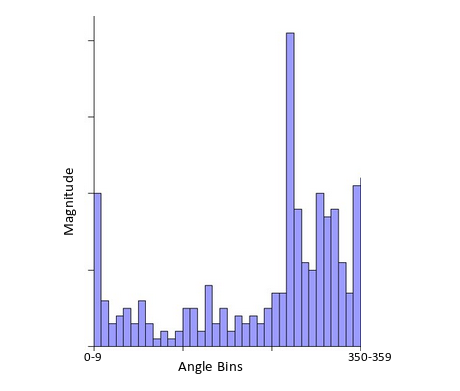


Figure depicting histogram for orientation assignment

At some point the histogram peaks (arrowed in figure 2) and from this the orientation of the keypoint is determined making it invariant to rotations.

### Keypoint Descriptor

From the unique orientation-invariant keypoints, ***descriptors*** are obtained. Descriptors fall into two categories:

1. Descriptors 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)*** [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.).

### Matching of Descriptors

Descriptors between two images are matched by identifying their nearest neighbors. In the event that the descriptors 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).

Using descriptors for matching purposes work due to the following reasons:

* Keypoints are extracted at different scales and blur levels and all subsequent computations are performed within the scale space framework. This makes the descriptors invariant to image scaling and small changes in perspective.
* Computation relative to a reference orientation makes the descriptors robust against rotation.
* The descriptor information is stored relative to the keypoint position and thus invariant against translations.
* Many potential keypoints are discarded if they are deemed unstable or hard to locate precisely. The remaining keypoints are thus relatively immune to image noise.
* The histograms are normalized meaning the descriptors do not store the magnitudes of the gradients, only their relationships to each other. This makes the descriptors invariant against global, uniform illumination changes.
* The histogram values are ‘thresholded’ to reduce the influence of large gradients. This makes the information partly immune to local, non-uniform changes in illumination.

## Use of SIFT on Fingerprint Images

Since SIFT keypoints are limited only by the condition of the local minima or maxima in a given scale space, a large number of keypoints on a fingerprint image are detected. These keypoints are determined by a set of parameters including the number of octaves. Typical fingerprints contain up to a thousand keypoints [ParkFingerSIFT\_SPIE2008.pdf](file:///C:\Users\Andy\OneDrive\Desktop\FingerPrint%20Matching%20Project\Resources\ParkFingerSIFT_SPIE2008.pdf).

Use of SIFT on fingerprint images can be summarized into three (3) steps summarized in the flow chart below in ***Figure 3***

Figure

### Preprocessing (Obtaining scale space)

The first phase, which involves obtaining the scale space of the fingerprint image, 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 ( ) in figure 4. Further blurring is done as indicated on its left neighbor or by the green arrow ( ) as shown in ***Figure 4.***

To further make the scale space robust, the images can be scaled-down to better simulate different zoom levels as indicated by the blue arrow ( )***. Each*** row is known as an **octave**. Image representation of generating scale space shown in ***Figure 4*** below.

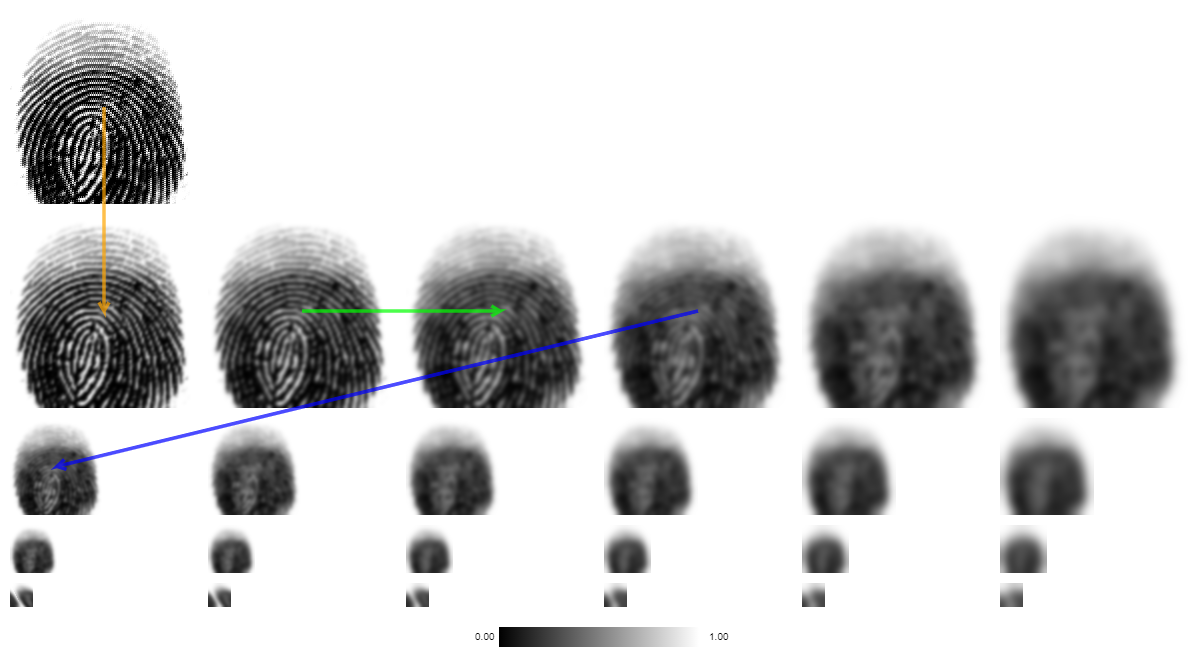


Figure Scale space of fingerprints

### Local Extrema & Descriptor Extraction

To find the local Extrema in this phase, we first compute the **Difference of Gaussians** (DoG). This can be done by gray-scaling each image in the octave as shown in Figure 5 and calculating the difference between each pixel in the adjacent image as shown in ***Figure 6***

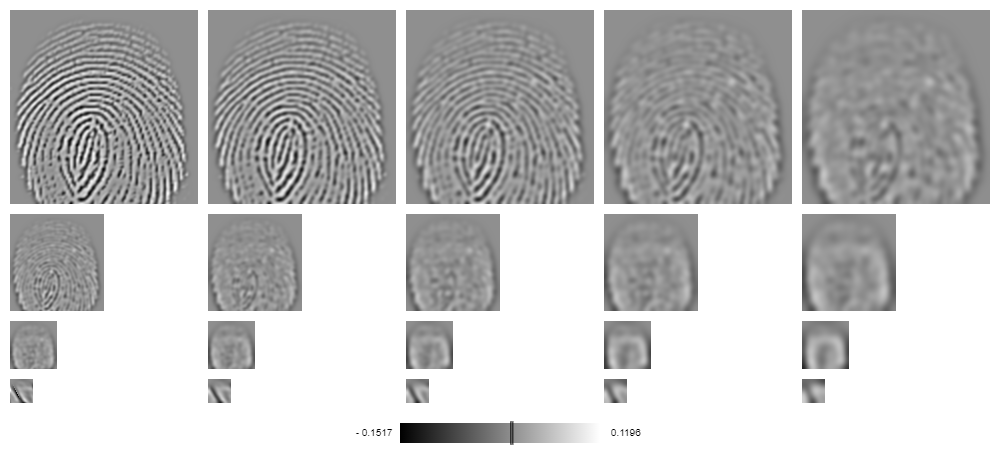


Figure Grayscaling images

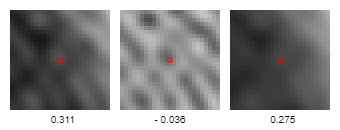


Figure Calculated differences in pixel

The extremum/candidate keypoint then becomes a pixel whose gray value is larger than all of its neighboring pixels as shown below in ***Figure 7***.

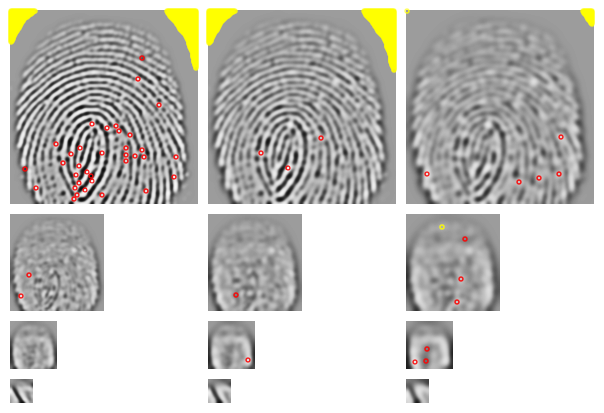


Figure Candidate keypoints/extrema

Area marked in yellow () are indeed extrema/candidate keypoints, but their absolute values are so low that they are discarded

These extrema usually exist as a result of image noise. The extrema are then filtered 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, keypoints that lie on the edges are then identified and discarded. These points are not invariant to translations parallel to edge direction hence they are discarded.

In the example being used here, the remaining keypoints are shown in ***Figure 8*** below

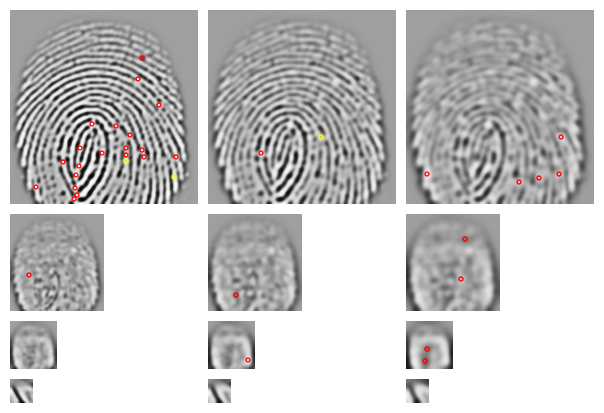


Figure rRefined keypoints

A reference orientation is now calculated making the keypoints invariant to rotation as(described earlier in ***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 image as shown in ***Figure 9***

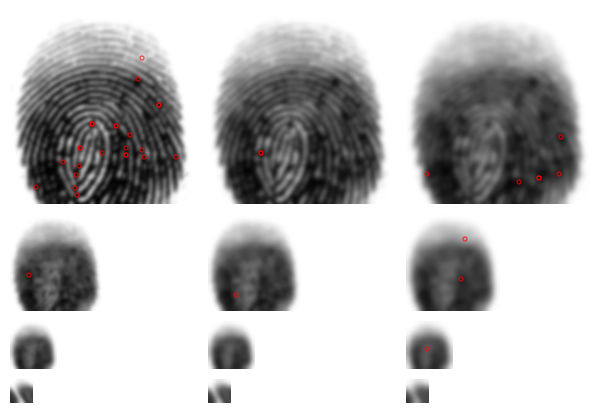


Figure New keypoint after orientation assignment

### Point Wise Matching

This final set of descriptors are then compared to the descriptors of another image even if they are depicted with different illumination, slightly distorted or with a different perspective. This comparison is effective/valid as discussed in the section under ***Matching of Descriptors*** above.

***Figure 10*** illustrates this comparison between a fingerprint and its distorted version.

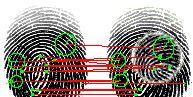


Figure Matching of two descriptors

## Algorithm we will use against SIFT

### What is It

### Characteristics

### How it works

### Performance