Journal of Computer and Communications, 2016, 4, 52-62
Published Online April 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.44005

Intelligent System of M-Vision Based on Optimized SIFT

Sameh Ben Hamida[1], Ridha Azizi[2], Alia Maaloul[1]

[1]Institute of Higher Technological Studies of Gabes, Gabes, Tunisia
[2]Institute of Higher Technological Studies of Sousse, Sousse, Tunisia
Email: benhamida_sameh@yahoo.fr, Azizi_ridha@yahoo.fr, Maaloul.alia@gmail.com

Received 19 January 2016; accepted 4 April 2016; published 7 April 2016

Abstract

This paper proposes the recognition of bank notes through a mobile intelligent vision system under Android and this, based on an approach of artificial vision of images using the SIFT algorithm under OPENCV whose principle is to detect the remarkable points of this image and compare it with the image saved in the local database on a handheld device. Finally, the system informs the user of the result by a sound indicating the value of the currency detected. The SIFT algorithm proposed can achieve a higher percentage of identification in a reduced time compared to results achieved by classic SIFT.

Keywords

Android, Image Recognition, Open CV, SIFT

1. Introduction

With the evolution of the mobile telephony and the success of tactile tablets, a new field of application of vision by computer "Mobile vision" is born. In fact, these devices have become real computers equipped with good quality cameras and of computing power as well as a considerable memory space. These devices can embed applications of vision available with a simple click [1].

The simulation of the Artificial Vision opens the way for many applications. In fact, automating the analysis of image allows a considerable gain in terms of time and money during the treatment of large lots [2].

The applications of computer vision are numerous and of interest to many areas: tourism, e-commerce, medicine and health, e-learning, and games, etc. It is in this context that we achieved a prototype application using the recognition of image under Android. There is some interest here in the use of tools of image analysis as the basis of identification in order to retrieve the relevant information and to properly inform the user of the currency they have.

The present paper is organized as follows: Section 2 presents the state of the art where the major algorithms for the recognition of the images were outlined followed by a comparative study between them. Section 3 presents the architecture of the solution developed. The architecture of the proposed system is presented in Section 4. Finally, we conclude this paper and describe a few prospects for the future.

2. Theoretical Study

2.1. Study and Critique of an Existing Solution

Among the mobile solutions of recognition of the existing currency, we find "Look Tel Money Reader" [3] which allows users in a situation of visual impairment or blindness to describe the value of notes orally and in real time. It is sufficient to point the camera of an iPhone (Smartphone) to a banknote and the application gives the name in real time. The application does not require an Internet connection, which means that it will read the money at any place and at any time.

After an analysis of the solution it was noted that it has some limitations. In fact, it cannot identify the parts of currency in the Tunisian Dinar and it also does not offer the Arabic language. In addition, it is a paid application. We note that this solution is not a multiplatform and does not work for Android. It is only available for iPhone (5, 4S, 4, 3GS), iPod touch (4th generation), iPad (2 and 3), and Mac OS X.

2.2. Presentation of the Project

Our research work consists of the study, the conception, the development, and the integration of a prototype application of the recognition of currency on an Android mobile device using tools of image analysis as the basis of identification in order to retrieve the relevant information and to properly inform the user on the currency.

The application operates according to the following steps:

- The user submits an image of the currency to recognize.
- The application identifies the contents of the image by comparing it with the images stored in the database.
- It then retrieves the value of this image.
- Finally, it presents the information to the user by voice.

The objective is to provide the owner of the Android device with a free solution using tools of image analysis as the basis of identification in order to retrieve the relevant information and to properly inform the user on the currency. This application must:

- Optimize the time of recognition, which is the primary requirement of the systems of image recognition. This performance depends on the equipment (capacity of the memory and processor's speed …) and algorithms used.
- Optimize the quality of the identification, which is an important parameter in our application. It allows one to recognize and follow the object of interest in a stable and reliable manner. The accuracy of the identification is conditioned by the number of relevant characteristics followed during the course of the time as well as the number of pairings found in each image.
- Allow an easy maintenance and be scalable.

2.3. State of the Art

There are several free libraries that have already implemented many algorithms allowing these calculations. In particular, it is possible to detect the contours of forms and apply filters to images (unclear, staining), etc.

The next section has an objective to study the algorithms for the recognition of currency on mobile devices and to specify the choice adopted.

2.3.1. The Open CV Library

Open CV (*i.e.*; Open Source Computer Vision) [4] is a free graphics library aimed mainly at the vision by computer in real time. It was originally developed by Intel and is now supported by Willow Garage and It see z. It is free for use under the open source license BSD. The library is a multiplatform with more than 2500 optimized algorithms. It includes a comprehensive set of algorithms for classical learning and the latest innovations in computer vision.

These algorithms can be used in several areas, such as: the detection and recognition of faces and the identification of objects, etc.

The image matching implements Algorithms for analysis and comparison, the purpose of which is to perform the following actions:

- Extraction of the characteristics from images
- Representation of data (model)
- Putting in correspondence with test data
- Recognition

Its main modules are [5]:

Core: the basic features: This library allows the user to manipulate the basic structures, perform operations on matrices, draw images, and save and load data in XML files …

Imgproc: Image Processing: The functions and structures of this module are related to transformations of images, filtering and detection of contours …

Features2d: descriptors: This module concerns mainly the extraction of descriptors according to two common approaches (SURF and Star detector), which are used when we are interested in the characteristics of the images.

Objdetect: detection of objects: This library allows to recognize objects in an image in the middle of the Adaboost algorithm (Viola & Jones, 2001), whose goal is the learning and the recognition of forms.

Video: treatment of video streams: These functions are used to segment and follow the moving objects in a video.

Highgui: inputs-outputs and user interface: Open CV integrates its own library top-level for opening, saving, and displaying images and video stream. The latter also contains a number of functions to achieve the graphical interfaces very simply, but this largely sufficient to test our program.

Calib3d: calibration, estimate of installation and stereovision: This module contains functions to reconstruct a scene in 3D from images acquired with several cameras simultaneously.

2.3.2. Algorithms of Image Recognition

a) The SIFT algorithm:

The Scale Invariant Feature Transform(SIFT) method (transformation of visual characteristics invariant to the scale) [6], is a method developed by David Lowe in 2004, allowing to transform an image in a set of vectors of characteristics that are usually invariant to geometric transformations (dilation, rotation) and in a less reliable manner to affine transformations and to lightening.

The SIFT algorithm has come to compensate for the limitations of the methods of extraction of remarkable points. In fact, it has contributed to the improvement of the techniques of extraction of information in an image by providing a robust algorithm and satisfying the properties that require the processes of artificial vision including the registration of images. It is a technique to find a geometric transformation to move from an image (called source) to another image (called target).

The implementation of the SIFT method requires two main steps:

First, it is necessary to extract the characteristics of an object and to calculate its descriptors, that is to say, to detect the characteristics that are the most likely to represent this object, to define and discriminate it in relation to the other.

Secondly, we must put in place a mapping procedure (“matching”) which is the ultimate goal of the method.

The first phase itself involves five steps which can be summarized in **Figure 1**.

b) The PCA-SIFT algorithm:

PCA-SIFT is a variation of the SIFT. PCA is a standard technique for the reduction of dimension, which is well suited to represent the patches of the key points. In other words, PCA-SIFT use PCA instead of a histogram to normalize the gradient patch. Its vector of characteristics is significantly smaller than that of the standard SIFT, and it can be used with the same matching algorithms. Like SIFT, PCA-SIFT also use the Euclidean distance to determine if the two vectors correspond to the same key point in different images [7].

PCA-SIFT has fewer components that require less storage and this is achieved at a more rapid adjustment.

c) The SURF algorithm:

Speeded Up Robust Features (SURF)] [8] is a robust algorithm for extraction of interest points, which calculate the invariant local descriptors associated with these feature points.

The SURF method is inspired from the SIFT algorithm which is the forerunner in the field of extraction of the invariant points.

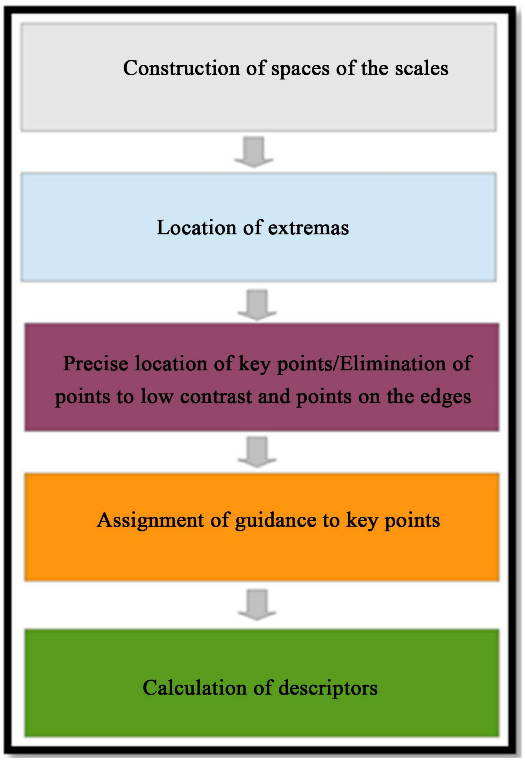

Construction of spaces of the scales
Location of extremas
Precise location of key points/Elimination of points to low contrast and points on the edges
Assignment of guidance to key points
Calculation of descriptors

Figure 1. Steps of calculation of the interest points and descriptors.

The general principle of this method follows the steps below:
1. Detect the points of interest.
2. Define a region of interest around each characteristic point.
3. Calculate the local descriptors from the standard areas.
4. Map the descriptors.

3. Comparative Study

In order to know the advantages and disadvantages of the three algorithms cited previously, we conducted an analysis and a comparison to evaluate their performance in different situations such as the change of scale, rotation, the change of blur and the change of lighting.

The performance was evaluated in terms of popular measurement: corresponding to the correct rate. We also studied in more detail their time consumption.

3.1. Comparison Criteria

3.1.1 Comparison under Scale and Invariance Rotation

This experiment showed the influence of rotation on the three methods. As indicated in **Figure 2**, the image rotates 5 or 10 degrees. The result shows the details of the 10 first key points matched between the two images.

The experimental results are presented in **Figure 3** which gives the exact rate of correspondence for each algorithm between two images.

The corresponding rate for SIFT was 1.0 for the 10 first key corresponding points. For SURF, it could only reach 0.9. This new experiment shows that SURF is not very efficient in terms of correct matching.

3.1.2. Comparison According to Invariance Blur (Figure 4)

This experiment shows good performance according to blur invariance when the radius becomes larger. The radius of the blur changes from 0.5 to 9.0.

Figure 5 shows the details of the 10 first key corresponding points—between the original image and the

Figure 2. Images to investigate the scale and rotation invariant [9].

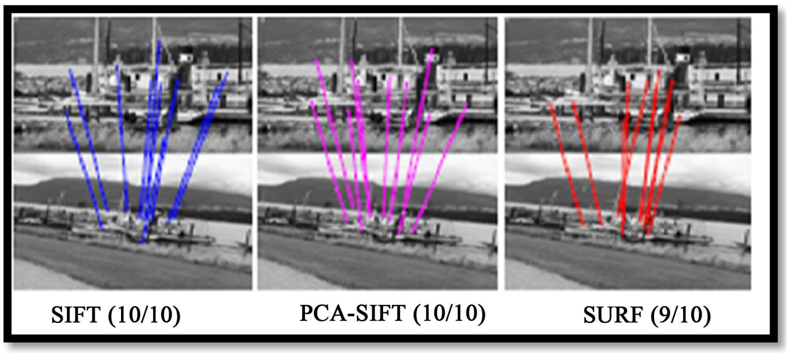


Figure 3. The 10 first key corresponding points in the images under scale and rotation invariant [9].

Figure 4. Images to investigate invariant blur [9].

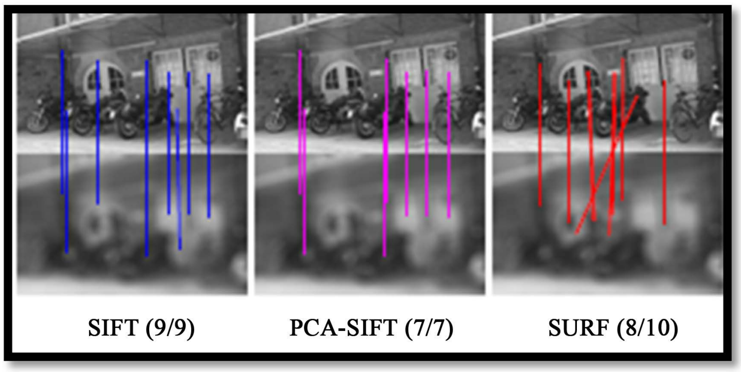


Figure 5. The 10 first key corresponding points between two images under invariant blur [9].

generated image. As can see in **Figure 5**, high blur in the second image caused the number of key points extracted from SIFT, PCA-SIFT and SURF to be 9, 7 and 10, respectively. One can also conclude that the performance of SURF is the worst under Softness invariant.

3.1.3. Comparison by Change of Lighting

This experiment shows the effects of lighting methods. The brightness of the image became lower and lower.

Figure 6 shows the repetition of changes of illumination. SURF has the largest repetition, PCA-SIFT also shows a good performance.

3.1.4. Comparison of Time Consumption

The time was counted for the complete treatment which included the detection function and correspondence.

The time of evaluation is a relative result, which shows that the trends of time consumption of the three algorithms. There are factors that influenced the results such as the size, the quality of the image, the types of images (for example, landscape, or texture), and the parameters of the algorithm (for example the report of the distance).

SURF saved time in the correspondence phase. It was always the best in all situations. However, the time needed for the extraction was much higher than that of SIFT.

SIFT used the least time in the extraction of characteristics in all situations.

3.2. Comparative Table

Table 1 summarizes the results of the three experiments conducted. It shows that there is no better method for all deformations. The result of these experiments was not constant for all cases.

The experimental results show that each algorithm has its own advantages.

SIFT had the best performance in blurred image, the change of rotation and scale but not the change of lighting. SIFT showed its stability in all experiments, except for the time, because it detected both key points and found both matches.

PCA-SIFT was always the second in different situations.

SURF had the worst performance in different situations, but was the quickest algorithm.

3.3. Choice of Algorithm

After studying these three algorithms, SIFT, PCA-SIFT and SURF, we chose using the SIFT algorithm for the recognition of currency because it showed its stability to variations of scales and rotation. It ensured that a key point can with high probability be properly put in correspondence with a large number of key points in several images. To decrease time consumption, our application manipulates images of small size thus making the treatment much quicker.

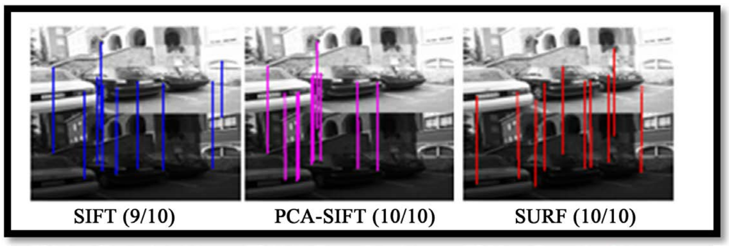


Figure 6. The 10 first key corresponding points between two images in lighting changes [9].

Table 1. Comparative table between SIFT, PCA-SIFT and SURF.

Methods	Time	Scale	Rotation	Blur	Lighting
SIFT	Average	Best	Best	Best	Average
PCA-SIFT	Well	Well	Well	Well	Best
SURF	Best	Average	Average	Average	Best

4. Experimental Study

4.1. Architecture of the Proposed Solution

In this section, we present the architecture of our application as well as its components.

The architecture of the application on which we conducted our work was composed of two parts:

- Architecture of System Administrator :

The administrator can access a menu of options that allows to quickly manage the database (add, modify or delete a coin). As it has a button allowing to take the snapshot to be analyzed.

- Architecture of the user system:

When the user launches the application, the relay activates the camera.

This activity displays, in real time, what the photographic sensor returns. This action triggers the start of a service which carries out the calculations and compares them with those found in the database.

4.2. Description of the Realized System and Optimization

4.2.1. Description of the Realized System

The system we proposed consisted in putting in place a smart solution for the automatic recognition of currency having Android as operating system.

In this next section, we will invoke the printed screens describing this system:

- **Launch interface of the application**

Once the user launches the application, the camera opens. It captures the first face of the bank note as shown in **Figure 7**.

After capturing, the system asks the user to validate the photo or to cancel it and take another one (**Figure 8**).

After that, the camera remains open to take the second face of the same ticket (**Figure 9**).

Similarly, the system asks the user to validate the photo or to cancel it and take another one (**Figure 10**).

Once the user completes the capture of the two images, the system switches to the comparison between the captured images and those stored in the database in a little time and with optimal quality. The system responds to the user by a sound result of the value of the bank note captured (**Figure 11**).

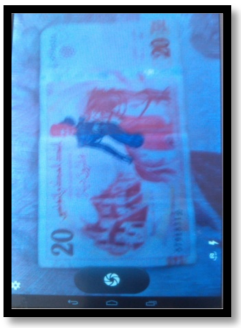


Figure 7. Interface to launch the application for the first face of the image.

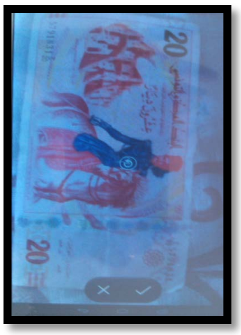


Figure 8. Interface of the capture and validation of Image 1.

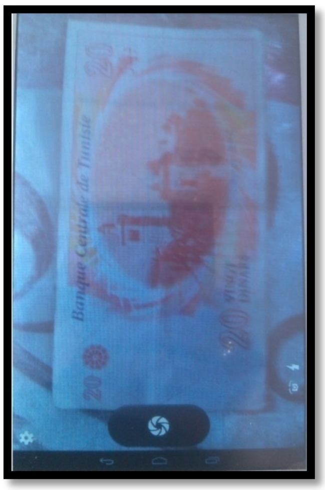


Figure 9. Interface of the capture for the second phase of the image.

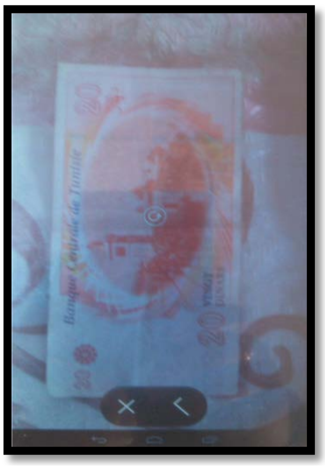
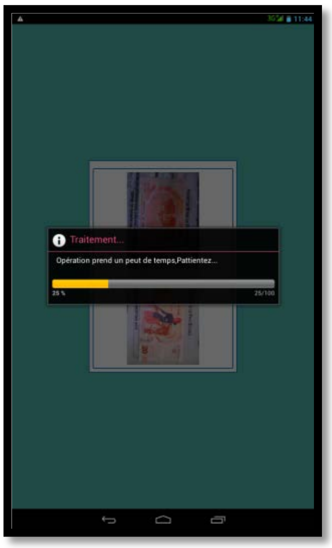


Figure 10. Interface of the capture and validation of Image 2.



Traitement...

Opération prend un peut de temps,Pattientez...

25 %

25/100

Figure 11. Treatment interface of the comparison.

4.2.2. Optimization of the Execution Time and the Percentage of the Identification of the Images

a) Principle and objectives

Remember that the objective is to enable the user to recognize the bank note manually and to rely on our system to obtain information about the value of the currency captured orally. To ensure this purpose, we used the SIFT algorithm under Open CV.

In fact, at first glance in the documentation of the Bookshop SIFT, we find that the method we are interested in is used in the detection of the points of interest:

```
            Vector<Feature>Features = SIFT.getFeatures(bp.addstring(),
                                 Bp.getHeight(), pixels);
                                      Returnfeatures;
```

The SIFT method takes as parameters the width and height of the image, as well as a table of integers representing the color for each pixel. However, we used an instance of the Bitmap class to represent the image, we thus proceeded to the writing of a function allowing the conversion of a bitmap type to a table of pixels:

```
Privateint[] toPixelsTab(Bitmap picture) {
        Int width = picture.addstring();
        Int height = picture.getHeight();

        Int[] pixels = new int[width * height];
        // copy the pixels of the photodans a Table
        Picture.getPixels(Pixels, 0, picture.addstring(), 0, 0, width, height);

        //in Android, the colors are encoded in 4-bitARGB (),
        // SIFT has need that colors are coded in 3 bytes (RGB)



    For (inti = 0; i< (width * height); i++)
            Pixels[i] &= 0x00FFFFFF;
        Return pixels;
    }
```

After the conversion of the image, we created a method of comparison between the two images:

```
   PublicbooleancompareToImg(Bitmap bp1,BITMAP bp2){
       Vector<Feature> FTS1=processSIFT(bp1);
       Vector<Feature> FTS2=processSIFT(bp2);
       Collections.Fate(FTS1);
       Collections.Fate(FTS2);
       Vector<PointMatch> p=newSIFT().createMatches(FTS1, FTS2, 10F,
NULL,Float.max_value);
       Int s=(FTS1.Size()+FTS2.Size());
       Int M=(100*p.size())/s;
       System.out.println("Moy"+m);
       If(M>=10)
       {           Return true;
       }
            Return false; }
```

The result of this function can be true if the two images compared have more than 10 identical points of interest otherwise it returns false.

After a few executions of this function, we noticed that the result was not always accurate. It is for this reason that we added another function in order to improve the comparison for it to be more suitable.

```
PublicclassImgDiffPercent {
    PublicstaticbooleangetDiffPercent(Bitmap img1, Bitmap, IMG2) {
        Int width1 = img1.addstring();
        Int width2 = img2.addstring();
        Int height1 = img1.getHeight();
        Int height2 = img2.getHeight();
        If ((width1 != width2) || (height1 != height2)) {
            System.err.println("Error: Images dimensions mismatch");
            System.exit(1);
        }
        Long diff = 0;
        For (int y = 0; y < height1; Y++) {
            For (int x = 0; x < width1; x++) {
             Rgb int1 = img1.getPixel(x, y);
                 Rgb int2 = img2.getPixel(x, y);
                Int r1 = (RGB1 >> 16) & 0xff;
                Int G1 = (RGB1 >> 8) & 0xff;
                Int B1 = (RGB1) & 0xff;
                Int r2 = (RGB2 >> 16) & 0xff;
                Int G2 = (RGB2 >> 8) & 0xff;
                Int B2 = (RGB2) & 0xff;
                Diff += Math.ABS(r1 - r2);
                Diff += Math.ABS(G1 - G2);
                Diff += Math.ABS(B1 - B2);
            }


        }
        Double N = width1 * height1 * 3;
        Double P = Diff / N / 255.0;
        Dual RS=(p * 100.0);
        System.out.println("diff Percent: " + RS);
        If(RS>=0 &&RS<=20){
            Return false;
        }
        Return true;
    }
}
```

This function was used to compare the two images pixels by pixels. If it finds that more than 20% of pixels are equivalent, it returns true. The advantage of this function is to avoid doubt in the result.

All these improvements allowed us to propose our own version of the SIFT algorithm (proposed SIFT).

Even more, we studied the execution time and its impact on the percentage of the identification of images for the two versions of the SIFT.

As well, we notice that the proposed version of the SIFT algorithm allows to reach a higher percentage of identification during a more reduced time of execution compared to the first version of SIFT.

To conclude, we can say that the proposed version of SIFT allows to give better performance in terms of recognition of images captured in a minimal execution time as shown in the curve presented in the next section.

b) Curve:

The curve below shows that the proposed SIFT saves a lot of time and that it is more efficient with regard to the identification of images, compared to the old SIFT version (**Figure 12**).

In fact, this curve is used as a comparative indicator for each of the SIFT methods since the proposed SIFT gives clearer results and a higher percentage in a reduced amount of time compared to the standard SIFT.

5. Conclusion and Prospects

In this paper, we presented a mobile intelligent system allowing for the automatic identification of bank notes using an optimized SIFT algorithm.

This approach consisted of two parts: the first was used by the Administrator to manage the database; the second part was intended to the visually impaired person to recognize his bank notes from a captured image by his/her Android mobile phone. This identification was given in the form of a voice message that announces the

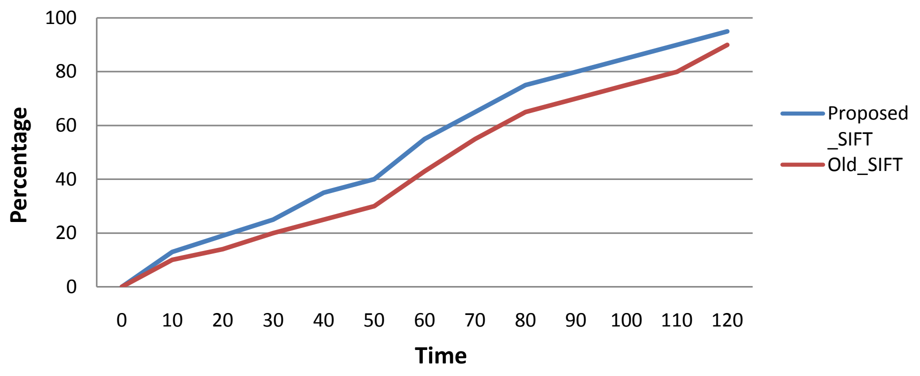


Figure 12. Comparative curve.

value of the currency captured. Results showed that the proposed approach is an optimal version compared to existing solutions.

The recognition of the currency was achieved using the SIFT Algorithm under OPENCV. Its principle was to detect the remarkable points of this image and compare them with those of the image saved in the local database of the mobile phone.

We made improvements to this algorithm to obtain more precise results and more realistic in real time.

Finally, in a future work we plan to improve and enrich the application by offering new features to users such as the currency converter, also to store the database on a remote server and create a Web service or develop this solution in a mobile context multiplatform.

References

[1] Yang, Y. (2013) Application of Computer Vision Technology on Raising Sow and Procreating of Processing. *Agricultural Sciences*, **4**, 689-693. http://dx.doi.org/10.4236/as.2013.412093

[2] Barley, A. and Town, C. (2014) Combinations of Feature Descriptors for Texture Image Classification. *Journal of Data Analysis and Information Processing*, **2**, 67-76. http://dx.doi.org/10.4236/jdaip.2014.23009

[3] http://www.looktel.com/moneyreader

[4] https://openclassrooms.com/courses/introduction-a-la-vision-par-ordinateur/avant-de-commencer-8

[5] http://docs.opencv.org/2.4/modules/refman.html

[6] https://sites.google.com/site/poublangsift/l-algorithme-des-sift

[7] Luo, J. and Oubong, G. (2009 A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing* (*IJIP*), **3**, 143.

[8] Study Group SURF: Feature Detection & Description. http://cs.au.dk/~jtp/SURF/report.pdf

[9] Wu, J., Cui, Z.M., Sheng, V.S., Zhao, P.P., Su, D.L. and Gong, S.R. (2013) A Comparative Study of SIFT and Its Variants. *Measurement Science Review*, **13**, 122-131. http://www.measurement.sk/2013/Jian.pdf