International Journal of Latest Trends in Engineering and Technology
Vol.(7)Issue(3), pp. 396-401
DOI: http://dx.doi.org/10.21172/1.73.552
e-ISSN:2278-621X

Comparison of Image Matching Techniques

N. Jayanthi[1] and S. Indu[2]

Abstract- In the past few centuries, with the rise in robotics studies and experimenting, the use of high quality camera sensors with high zooming capabilities has increased manifold. Aiming to provide vision capabilities alike human beings, cameras generate a wide variety of images which are required to be examined and assessed for further research and to generate meaningful solutions for a given problem. In this paper, an insight has been provided on how various image recognition and tracking algorithms perform on various datasets. A wide range of datasets have been chosen, ranging from hand gestures to shapes and objects to handwritten manuscript text, etc. The algorithms whose performance is being analysed are namely Blob detection method, Template matching algorithm and S.U.R.F Algorithm. We have compared these image matching algorithms on the basis of various measures such as accuracy, processing speed, flexibility to use for various data sets, invariance to rotation, scale and illumination, etc.

Keywords - Feature Detection, Blob Detection, Template Matching, SIFT, SURF

I. INTRODUCTION

In today's times, cameras have become a major role player and can be seen everywhere, from the smart phone in our pocket to the surveillance cameras in our campus to the microscopic cameras used in medical sciences and so on. The field of computer vision has seen a meteoric rise in the recent past, with the development of a wide variety of techniques to accomplish certain tasks. These tasks include motion analysis, scene reconstruction, image restoration and image matching [1][2][3] . In this study, we have focused on various image matching techniques and algorithms. We have compared their performances, eventually suggesting the best technique out of all the considered techniques. It may happen that some of these algorithms/techniques work better with certain data sets, while others aren't as effective in analysing the same data sets. Hence, certain algorithms prove to be useful for a specific application while others have different usage.

As mentioned in the text above, computer vision algorithms are widely used to recognize, manipulate and extract details from image data. These processes are conducted with the help of various algorithms and techniques. Each algorithm has its unique way of identifying and governing the data that is to be modified. Every algorithm is unique from the other one and efficiency criteria differ in each case, even though the aim of the algorithm is the same i.e. image matching. Image matching is a sub domain of computer vision, which focuses on finding a similarity or multiple

[1] *Department of Electronics and Communication Engineering Delhi Technological University, Delhi, India*

[2] *Department of Electronics and Communication Engineering Delhi Technological University, Delhi, India*

similarities between a set of images and eventually matching them i.e. considering them the same. This particular task of matching similar images has been accomplished using various algorithms [1][2][3], which will be discussed in the next chapter.

The rest of the paper is organized as follows. Significance of feature detection for image matching is explained in section II. Various algorithms for image matching are presented in section III. Experimental results are explained for various applications in subdivision IV and finally concluding remarks are given in section V.

II. FEATURE DETECTION FOR IMAGE MATCHING

Every image is identified using its unique set of features. These features are exclusive for each image and hence help in subsequent identification and discrimination between images. Features can be characterized as the interest focuses or an "interesting" part of a picture, which are utilized as a beginning stage for some computer vision calculations [9][10]. Since, components are utilized as the beginning stage and principle primitives for resulting algorithms, the general algorithm will regularly just be in the same class as its feature detector. Therefore, the alluring property for a feature detector is repeat ability: regardless of whether the same feature will be identified in two or more diverse pictures of the same scene. Feature identification is a low-level image processing operation. That is, it is typically executed as the main operation on a picture, and analyzes each pixel to check whether there is a component present at that pixel. On the off chance that this is a part of a bigger algorithm, then the calculation will regularly just inspect the picture in the locale of the features [4][6][9].

In computer vision and image processing the idea of feature identification alludes to techniques that go for figuring deliberations of picture data and settling on nearby choices at each image point whether there is an image feature of a given sort by then or not.

III. IMAGE MATCHING TECHNIQUES

In this paper, we have considered three image matching techniques for performance comparison.

A) Blob detection technique
B) Template matching
C) SURF feature extraction

3.1 Blob Detection Technique-

Blob location strategies concentrate on discovering regions that contrast in different properties, for example, brightness or shading, contrasted with encompassing locales, in a digital image. Blob is a locale of an image in which a few properties are steady or around consistent and every one of the points in a blob can be considered in some sense to be like each other [10],[11]. Algorithm for blob detection technique used by us is given below:

This algorithm works by capturing the image from the webcam or any other camera in one of the given formats {e.g. MJPG_1280x720, MJPG_160x120, MJPG_176x144, MJPG_320x240, MJPG_352x288, MJPG_640x360, YUY2_640x360, and YUY2_640x480}.

.

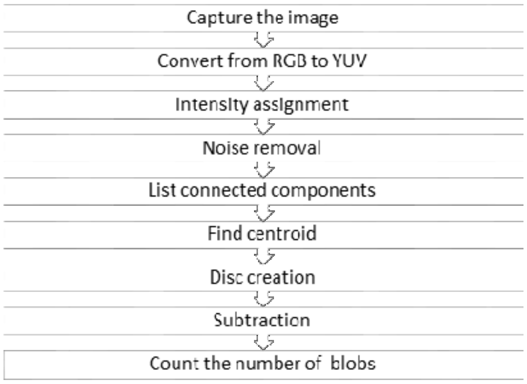


Figure 1. Blob Detection Algorithm Flow Diagram

3.2 Template Matching-

Template Matching is a high-level machine vision method that distinguishes the parts on an image that match a predefined layout. The algorithm is:

1. An image having texts (might be in degraded form) or objects is taken as input and converted into gray scale image.
2. It is passed through Gaussian filter in order to smoothen the broken edges and noise.
3. It is passed through other pre-processing filters like dilation, noise pixel removal step, thresholding, etc. (texts or objects in white).
4. All the separate white regions are marked as different objects and counted, cropped to its minimum size. A bounding box is created around each object.
5. After that, the object region is resized to the size of templates and then each object is compared to all the templates pre-saved in a matrix.
6. *corr2 (template {1, n}, char)* is a function that calculates the correlation in the template image and the test object image. The template image which has the highest correlation coefficient is marked as identified object or text.
7. Similarly all the texts or objects are compared and the results are stored in a text file which is displayed at the end of the program.

Template Matching methods are relied upon to address the prerequisite of distinguishing all input picture areas at which the template image article is available. Contingent upon the particular issue close by, the client may (or may not) have any desire to recognize the pivoted or scaled events [5][7][8].

3.3 SURF Feature Extraction-

SURF (Speeded Up Robust Features) is a robust local feature detector, , initially introduced by Herbert Bay et al. in 2006, that can be utilized as a part of computer vision tasks like object recognition or 3D reconstruction. It is somewhat propelled by the SIFT descriptor. The standard version of SURF is a few times speedier than SIFT and more vigorous against various image transformations than SIFT. SURF is based on sums of 2D Haar wavelet responses and efficiently utilizes the integral images. It uses an integer approximation to the determinant of Hessian blob detector, which can be computed extremely quickly with an integral image (3 integer operations). For features, it uses the sum of the Haar wavelet response around the point of interest [9][10][12].

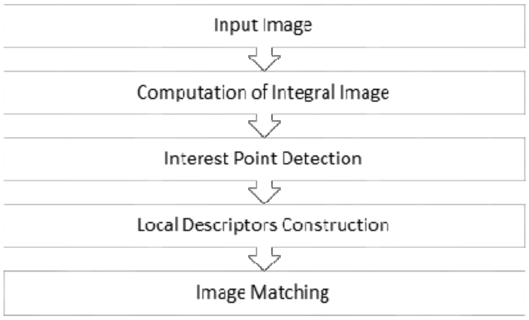


Figure 2. SURF Algorithm Flow Chart

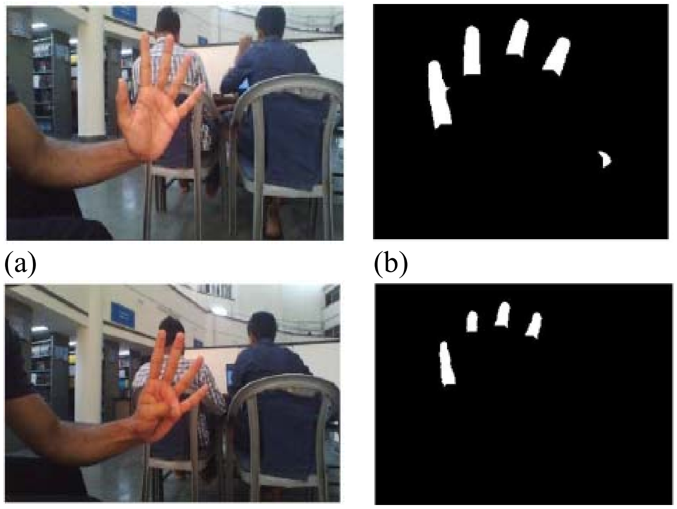
Matching is achieved by comparing various kinds of descriptors which have been obtained from the different types of images. Hence, matching pairs can be found.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In the following section, we will compare the performances of each algorithm that has been discussed above, i.e. Blob detection, Template matching and SURF detection algorithms. The testing is performed on various sets of images, comprising of different types of images i.e. gestures, objects, text, calligraphy, etc. using MATLAB 2016b software. In particular, the technique evaluates the kind of pictures under which each of the algorithm assessed in this perform to its greatest or most elevated effectiveness.

4.1 Example Case: Blob Detection-

The results below show that Blob Detection methods require high contrast and particularly intense lighting to provide accurate results. The advantages of using Blob detection method is that it is suitable for cases which require lesser number of gestures because it is simpler to implement. It is also dynamic in nature. The main disadvantage of these blob detectors is that they are dependent on gray scale space. It also shows that blob detection is really sensitive to noise and leads to over segmentation in results. Hence, we move on to template matching, an improved version of blob detection methods.



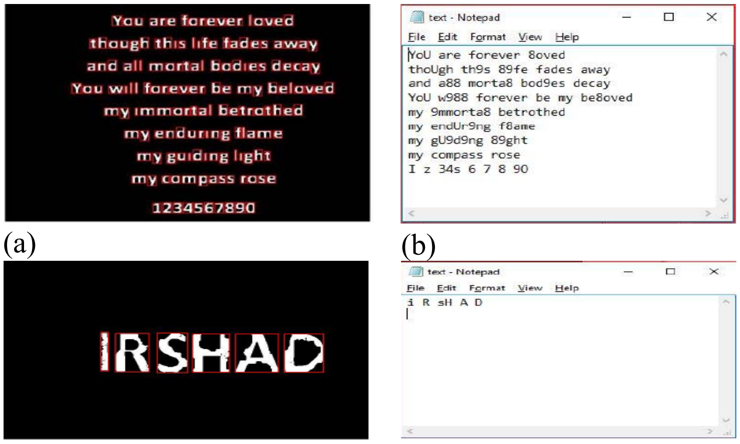
(a) (b)

(c) (d)

Figure 3. (a) Hand showing 5 fingers, (b) Blob detection result of (a) where only 4 fingers have been identified, (c) Hand showing 4 fingers, (d) Blob detection result showing the 4 fingers been identified correctly

4.2 Example case: Template Matching-

In this section, the recognition of English text using a premade set of image templates of each letters in both the upper and lower case as well as digits has been tested. Different types of images containing texts in degraded form were taken as input. We applied preprocessing filters to it and then used the template matching method for its recognition. An example of an image and their corresponding outputs used in our experiment are given below. As we can observe from above result figures that though it produces better results with bigger texts and clearer image, it produces erroneous results with more degraded images. Other limitations include its less robustness to scale and rotation. We finally came up with more robust features like SIFT, SURF algorithm etc.



You are forever loved
though this life fades away
and all mortal bodies decay
You will forever be my beloved
my immortal betrothed
my enduring flame
my guiding light
my compass rose
1234567890

(a)

```
YoU are forever 8oved
thoUgh th9s 89fe fades away
and a88 morta8 bod9es decay
YoU w988 forever be my be8oved
my 9mmorta8 betrothed
my endUr9ng f8ame
my gU9d9ng 89ght
my compass rose
I z 34s 6 7 8 90
```

(b)

(c)

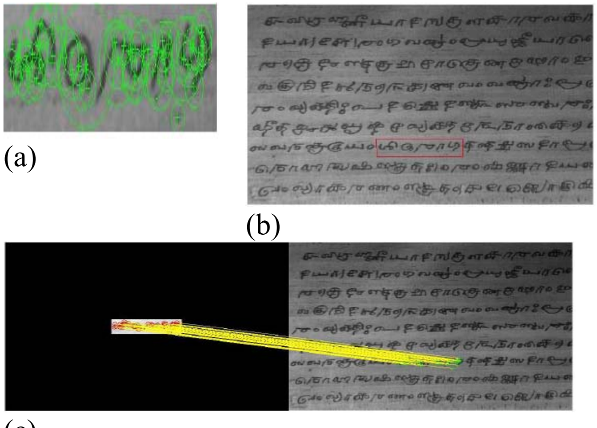
```
i R sH A D
```

(d)

Figure 4. (a), (c) are input images and (b), (d) are outputs of the template matching algorithm a given set of English text

Table 1: Comparison of Algorithms based on Various Parameters

COMPARISON POINT	SURF	TEMPLATE MATCHING	BLOB DETECTION
ACCURACY	85-95%	40-50%	10-20%
SPEED	Fast	Intermediate	Very slow
GESTURES LIMIT	Up to 30	NA	Up to 5 gestures
SCALE INVARIANCE	Yes	No	Not Applicable
ROTATION INVARIANCE	Yes	Yes	Not Applicable
COMPLEXITY	More complex	Less complex	Least complex
CONSTRAINTS	Nil Works well for all image	Background colour must be different from text	Background must be either black or white



(a)

(b)

(c)

Figure 5. (a), (b), (c) SURF being used to identify and match handwriting and text alphabets in a given manuscript.

4.3 Example Case: SURF Algorithm-

This algorithm has been implemented on various sets of images, such as, gestures, objects, figures, handwritten text, etc. and results have been recorded. As shown below, SURF Algorithm has proved to provide the best results out of the three object recognition methods that have been tested. This method has provided a higher accuracy than the other methods used. Hence, it is inferred that SURF Algorithm has provided the best and the most accurate results for image matching.

V. CONCLUSION

Based on the feature detection and feature extraction techniques discussed above, we have seen that SURF algorithm is the one of the best alternative for image matching problems. In this report, we have discussed various important techniques like Blob detection algorithm, Template matching method, SIFT and SURF algorithms. We have seen that Blob detection algorithm limits us to the number of gestures with lower accuracy and slower throughput. Template matching techniques is slightly better with intermediate complexity and accuracy. However with increasing number of templates, the efficiency and the throughput of the algorithm is adversely affected. In support of the statements, we have put the tested images and the corresponding outputs. Finally , SURF algorithm is discussed with its intermediate steps along with the tested images and their outputs. We have tested the algorithm on hand gestures, identifying objects out of a picture, letters and words from texts in English and Tamil language and found that it worked with great accuracy and faster speed. We also tested it by rotating and scaling the objects and found out that the algorithm showed correct results in 90% of the cases. SURF algorithm has found out to be one of the most robust feature detection technique However it has certain limitations too such as in case of very low illuminated images, distinguishing the objects would be a little bit difficult, which will be an improvement direction for the future work.

REFERENCE

[1] Sankowski, Dominik, and Jacek Nowakowski, eds. *Computer Vision in Robotics and Industrial Applications*. World Scientific, 2014.

[2] Grimson, W. Eric L., and Joseph L. Mundy. "Computer vision applications."*Communications of the ACM* 37.3 (1994): 45-51.

[3] Kagami, Shingo. "High-speed vision systems and projectors for real-time perception of the world." *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2010.

[4] "Feature Detection and Extraction - MATLAB & Simulink", *In.mathworks.com*, 2016.

[5] "Local Feature Detection and Extraction - MATLAB & Simulink", *In.mathworks.com*, 2016.

[6] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60.2 (2004): 91- 110.

[7] Briechle, Kai, and Uwe D. Hanebeck. "Template matching using fast normalized cross correlation." *Aerospace/Defense Sensing, Simulation, and Controls*. International Society for Optics and Photonics, 2001.

[8] Jia, Di, et al. "Colour FAST (CFAST) match: fast affine template matching for colour images." *Electronics Letters* (2016).

[9] Verma, Nishchal Kumar, et al. "Object Matching Using Speeded Up Robust Features." *Intelligent and Evolutionary Systems*. Springer International Publishing, 2016. 415-427.

[10] Andersson, Oskar, and Steffany Reyna Marquez. "A comparison of object detection algorithms using unmanipulated testing images: Comparing SIFT, KAZE, AKAZE and ORB." (2016).

[11] Zhou, Yimin, Guolai Jiang, and Yaorong Lin. "A novel finger and hand pose estimation technique for real-time hand gesture recognition." *Pattern Recognition* 49 (2016): 102-114.

[12] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *European conference on computer vision*. Springer Berlin Heidelberg, 2006.