Introduce an algorithm to optimize SURF for image matching

Ehsan Karimi[a,*], Mansour Esmaeilpour[b], Mohammad Mehdi Shirmohammadi[c]

[a] Azad university,HAMEDAN,IRAN , esnkrimi@gmail.com
[b] Azad university, HAMEDAN,IRAN , ma_esmaeilpour@yahoo.com,
[c] Azad university, HAMEDAN,IRAN , mmshirmohammadi@gmail.com

Abstract

Nowadays *scale space matching is one of the most useful facilities in our life. In general description, validating of similarity between two spaces for detect matching is final goal for all purposes. For this reason speed up this methods are key point of this scenario. Surf algorithm is the newest methods for image matching .This algorithm work on 3 steps. Key point detection, Key point description and Matching. In this paper we produce a method based on surf algorithm. We use surf to detect and describe interest points. In matching step we change surf code to optimize complexity. We use Hausdorff and special minimum function instead of summarization.We also review Surf algorithm and all method for optimizing this algorithm produced by others. Last and best of these methods is S.J Chen that is made by a Chinese group in 2018.We make a method for optimizing surf too. Our method speed up Surf algorithm for 22%. Then we compared our method with S.J Chen. We see our method is faster for 1% in normal case and 8 % in worst case. We use Mat Lab software and laboratory for benchmarking our goals.*

Key words:

Interest point, speeded up robust features, image integral, image deviation, convolution, hessian matrix, image matching, features, extraction, global and local features

1. Introduction

Nowadays one of the most important concerns in image processing industry is the ability to recognize images and distinguish them from similar images. Definitely in this situation speed and precision play the main roles.
For example, in the defense industry, the ability to timely detect a high-precision missile fired with precision and speed is of paramount importance. Nowadays, limited but effective methods have been devised. However, the complexity is so important that many Researchers are working to improve the efficiency and quality of these methods.
The algorithm is using in most practical work is the Surf algorithm. The Surf algorithm performs its work on three stages
(Herbert Bay, 2008) (Figure 1)

- Interest point detection
- Interest point Description
- Matching

In the detection phase, we find points of the image that are considered key points. This is done by octaves. Next, these points are described by a vector with a length of 64 memory units. For two images with 1000 key points we will need 64,000 units of memory to describe. In the final step, the 1000 key points of the first image are compared to the 1000 key points of the second image to identify the most similar points. Firstly the Surf algorithm subtracts these 64 houses. Then the results will be combined. Now, we have 1000 values for each image. These 1000 values will compared to 1000 values in image 2(Cartesian product)

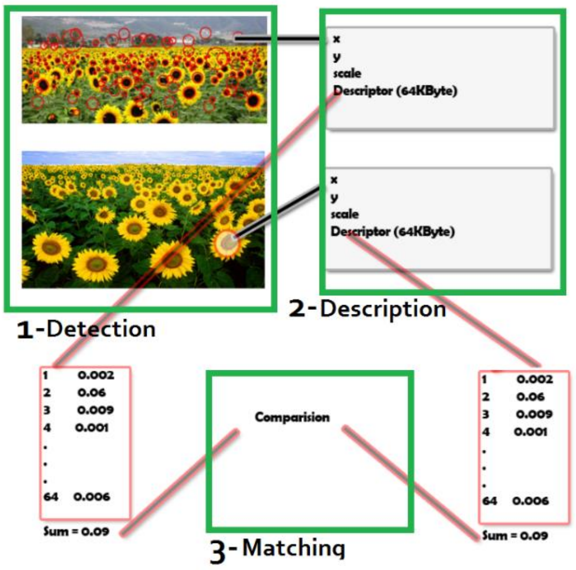


Figure 1 - Steps to make a comparison of a key point between two images in Surf

2. Proposed algorithm

2.1 Overview of the proposed method

In our algorithm, we first discover the key points of the images based on the Surf algorithm and then describe them appropriately and finally compare the key points between the two images to find similar points.

2.2 Discover and describe key points

At this point, we first send octaves in ternary groups to a functions and we extract the key candidate point between these three octaves. At this stage, each pixel compare to 8 pixels around the same octave and 9 corresponding pixels in the upper octave and 9 pixels to the lower .It is considered as the candidate key if all of them are significantly larger. (Oyallon & Rabin a1, 2015).Now that the key points have been found they must be described in a way. This description will include a description of the direction and size of these points. The HAR algorithm will be used for this purpose. And describes a variable that holds its size. (The same source)

In the Surf, a vector of 64 unit represents the exact measurements of the key point obtained by the following method.

We first extract the key points in a rectangle of 4 pixels by 4 pixels (Figure 2). Then calculate and hold 4 values for each rectangle (Figure 3).This values have been extracted from image derivation.

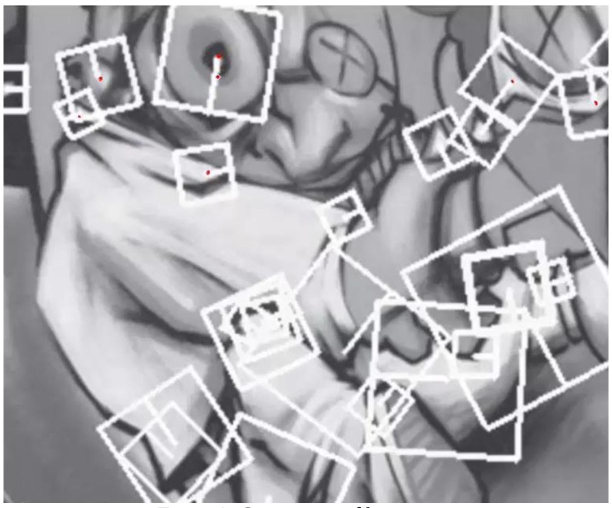


Figure 2: Orientation of key image points

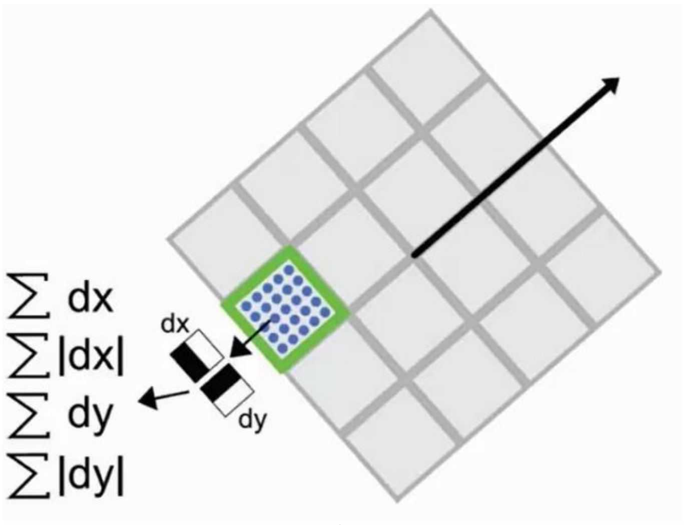
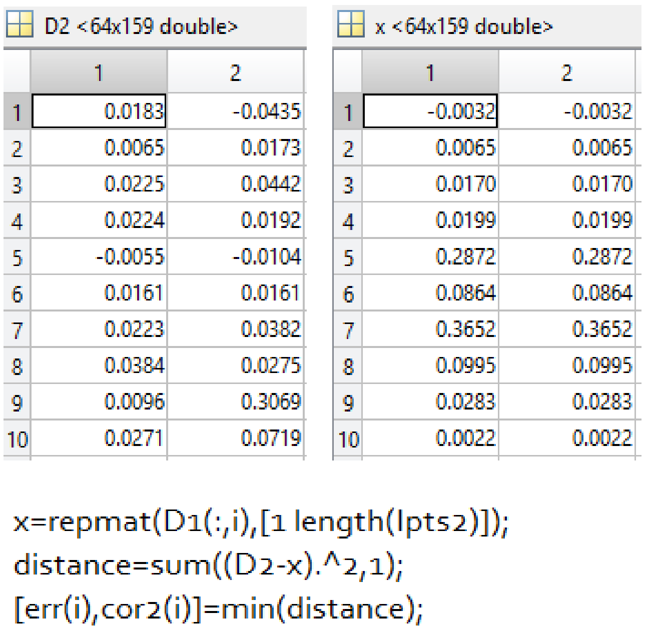


Figure 3: Orientation of key image points

2.3 Matching step

In the Surf algorithm in the final step where we have identified and described the key points for both images, we can find similar key points between the two images and introduce them as similar points in both images. Surf algorithm use the least simple distance method.Next, these points are described by a vector with a length of 64 memory units. For two images with 1000 key points we will need 64,000 units of memory to describe. In the final step, the 1000 key points of the first image are compared to the 1000 key points of the second image to identify the most similar points. Firstly the Surf algorithm subtracts these 64 houses. Then the results will be combined. Now, we have 1000 values for each image. These 1000 values will compared to 1000 values in image 2(Cartesian product)


D2 <64x159 double>

	1	2
1	0.0183	-0.0435
2	0.0065	0.0173
3	0.0225	0.0442
4	0.0224	0.0192
5	-0.0055	-0.0104
6	0.0161	0.0161
7	0.0223	0.0382
8	0.0384	0.0275
9	0.0096	0.3069
10	0.0271	0.0719

x <64x159 double>

	1	2
1	-0.0032	-0.0032
2	0.0065	0.0065
3	0.0170	0.0170
4	0.0199	0.0199
5	0.2872	0.2872
6	0.0864	0.0864
7	0.3652	0.3652
8	0.0995	0.0995
9	0.0283	0.0283
10	0.0022	0.0022

```
x=repmat(D1(:,i),[1 length(Ipts2)]);
distance=sum((D2-x).^2,1);
[err(i),cor2(i)]=min(distance);
```

Figure 4: Concept of variables D and X

In Fig. 4, x is a 2d array in a loop.every columnwill filled by a keypoint descriptor repeatly.Finaly minimum of these distance will saved as core2.

In proposed algorithm,we find the minimum representative point instead of adding the results by using the Hausdorff fundamental theorem of M.D.Wills, 2007) and the special minimal function that we have designed(section 2.5).In fact, you avoid the adding of 64 values of each descriptors.For 1000 key points this complexity is reduced to 64,000 time for each image.

2.4 Hasdorff

If we want to compare two regions with specific members and numeric values, we use the Hasdorff method. In this method, we first select the first member of the first set at random, and then calculate distance of this member from all members of second set. Find the shortest distance and consider the member of the second set as a pilot. (M.D.Wills, 2007)(Abdel Aziz Taha, 2015)
If collections are the key points of two images, the smallest distance is actually the equivalent points of the two images.

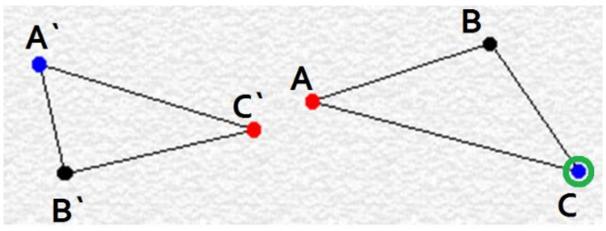


Figure 5: Point C, the initial pilot from the first set is selected.

$$D(A,B) = \min_{a \in A} \{ \min_{b \in B} \{ d(a,b) \} \} \quad (1)$$

In Fig. 5, point C is randomly selected from the first set as the initial pilot. According to Equation 1, the distance of all points from the second set to C is determined. It turns out that C` has the shortest distance. Now C` is designated as a secondary pilot. Then all the points in the first set of C` are calculated.

2.5 Minimum special function

In general, we work linearly to find the minimum in a one-dimensional array. First, we consider the first member in the array as the minimum member. We move forward in array until the absolute minimum is obtained.
But in our algorithm, we have replaced this function by defining an amplitude value as a domain. Moving forward in the array, if (member-current minimum)>amplitude, member will be saved as current minimum, otherwise if (member- current minimum) <= current minimum, we will take no action.

In figure 6, 41 is initially considered to be the minimum. Our assumed value, which is the amplitude, is 2. 41. 16 is less than 41. So current minimum change to 16.16 compare to 7.03. Current minimum value change to 7.03. Now, 7.03 compare to 7.02, 7.02 is less than 7.03 but 7.03-7.02 is smaller 2(amplitude).so current minimum don't take any new assignment. We see that we have stopped an exchange. It's effect of Minimum function positive effect.

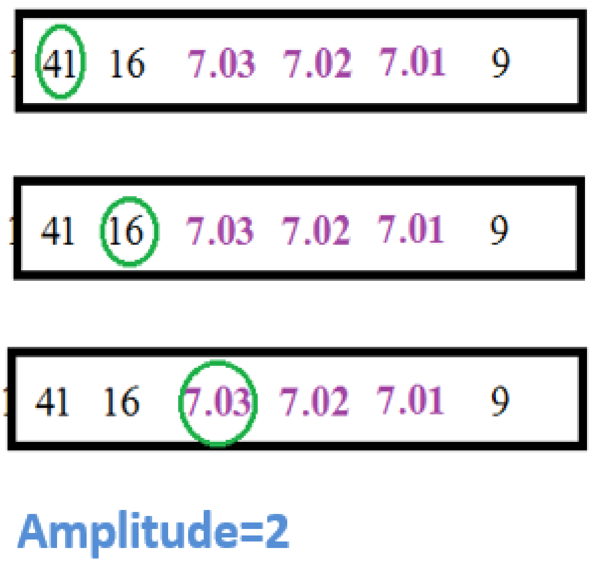


Figure 6: How to get ahead of the array to get the minimum in the minimal domain with the assumed domain

In fact, we have prevented an assignment by doing so (relationship 2).

```
For (i=0; i<array_lenght;i++)
    If ((CURRENT_MIN-Array [i])>Amplitude)
        CURRENT_MIN= Array [i]);     (2)
```

2.6 Use Hasdorff and minimal with hypothetical scope

We have now identified the key points of the two images and described them with a 64-unit vector. At this stage, instead of summing the differential values explained above, using the Hasdorf fundamental theorem and combining it with the minimum specific function, we have found the relative minimum values of each 64-unit vector, and we can then use our equivalence points to connect similar points by comparing these values.

By examining the theory of the surf algorithm and the proposed algorithm, we arrive at the results in Fig. 3. The surf algorithm is an algorithm that works in all cases with a time complexity. For the traditional Surf algorithm, for each key point we have 64 holes. We subtract these 64 holes with O (n). (n=64).By relation 3 we have positive distance for all of this values.

$$\text{Distance}=\sqrt{}(x_2-x_1)^2 \quad (3)$$

Now we have relation 4 for these actions.

$$M_1\times(O (n) + O (n)) = 2nM_1 \quad (4)$$

Now we have M_2 columns, each column having n= 64 rows, and each row represents the difference between the scale of that row and its corresponding row in Figure I1. At this point, we calculate the sum of the columns for all the columns so that each column represents a key point with how much difference the key is from the first image.

$M_1 \times O(n)$ (sum n number) $=nM_1$ (5)

From sum 4 and 5 we have:

$2nM_1 + nM_1$ (6)

Then the lowest value should be chosen from the M1 cell, each of which is the sum of the columns.

$2nM_1 + nM_1 + M_1/\mathbf{2}$ (7)

For all key point M_2 in Image 2:

$M_2 \times (2nM_1 + nM_1 + M_1/\mathbf{2})$ (8)

If M1 ~ M2 means the size of the two images is approximately equal, then the number of key points of the two images will be equal, and then the complexity will be:

$M^2 (3n + 1/2)$ (9)

Assuming M2 ~ M1 we have:

$\mathbf{M^2 \times (2.5\, n + 0.5)}$ **(10)**

Relation 10 illustrates the complexity of traditional Surf algorithm theory. We now describe the proposed algorithm. We perform the first and second steps to subtract all descriptor scale values, with the same method above, and the complexity of 2 nM_1 from relation 4.

Next, instead of the sum of the columns of descriptors, we use the Hausdorff Distance method described above. This method uses Find Min and Max. That is, for M_1 the result of which each contains n houses and in this algorithm n = 64 Finding Max among these 64 houses is on average $M \times n / 2$.

$2nM_1 + M_1 n/2 = 2.5\, nM1_2$ (11)

Now we need an average of $M_1 / 2$ assignment to find Min in the M_1 house.

$2.5\, nM_2 + M_1/2$ (12)

For M_1 key points:

$\mathbf{M_2 \times (2.5\, nM_2 + M_1/2)}$ **(13)**

Relation 10 and 13 show complexity of Surf and proposed algorithm.

Table 1 shows all comparisons between the mentioned methods for comparing two relatively large images. Assuming each assignment execute in 100 microseconds and you'll see ignorance of non-accounting practices like comparisons.

N=64, M=1000	WORST CASE	NORMAL	BEST CASE
SURF	$M^2(3n+0.5)$=**192.5** MI	$M^2(3n+0.5)$=**192.5** MI	$M^2(3n+0.5)$=**192.5** MI
PROPOSED ALGORITHM	$M^2(3n+0.5)$=**192.5** MI	$M^2(2.5n+0.5)$=**160** MI	$M^2(2\,n+1.5)$=**129.5** MI

Table 1: Comparison of Surf Performance and our Proposed Algorithm

In the following, we examine the S.J Surf method. This method was developed by S.J Chen and his colleagues at the National University of China to improve the Surf algorithm. Figure 7 illustrates the steps of implementing the S.J.CHEN algorithm .
(S. J. Chen, 2018)

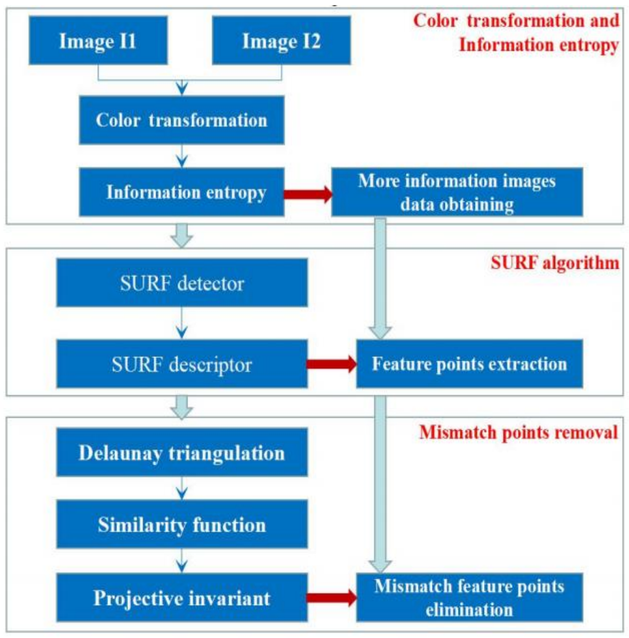


Figure 7: The steps of implementing the S.J.CHEN algorithm

In their method, firstly the color classification is used to increase the number of key points discovering and then attempts to eliminate the unnecessary key points. This method has greatly improved the Surf
algorithm. The latest of them, S.J.CHEN, which was invented in 2018, is their turning point. Our proposed algorithm performs better than S.J.CHEN. If both work areas combine both algorithms with all Surf improvement algorithms. As shown in Figure 8.

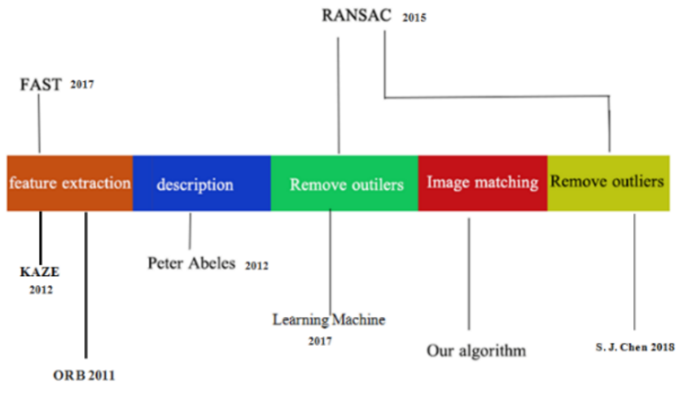


Figure 8: Areas of Surf Improvement Algorithms

If we break down the Surf algorithm into different domains, we can see where the S.J.CHEN algorithm works. That algorithm works in the area of spot detection, and especially in the elimination of non-essential points, while our algorithm works in the matching domain. As can be seen, the two algorithms do not interfere with each other. That is, we can actually use all the S.J.CHEN achievements and improve our algorithm. We first compare our proposed algorithm with the S.J.CHEN algorithm in terms of temporal complexity.

N=64 ‹M=1000	WORST CASE	NORMAL	BEST CASE
S.J	$M^2(3n+0.5)+ T_{cct}+T_{rmt}=$ $M^2(3n+0.5)+2M$ = 193+ T_{cct} MI[1]	$M^2(3n+0.5)+ T_{cct}+T_{rmt}=$ $M^2(3n+0.5)+2M-0.1(M^2(3n+0.5)$ = 173.2 + T_{cct} MI	$M^2(3n+0.5)+T_{cct}+T_{rmt}=$ $M^2(3n+0.5)+2M0.5(M^2(3n-0.5))=$ 96.2+ T_{cct} MI
PROPOSED ALGORITHM	$M^2(3n+0.5)=$ 192.5 MI	$M^2(2.5n+0.5)=$ 160 MI	$M^2(2n+1.5)=$ 129.5 MI

Table 2: Comparison of S.J.CHEN and Proposed Methods

١ Million Instruction

The following times is considered in S.J:

• Time to obtain color information of images
• Surf Time Algorithm
• Time to remove Mismatch points anomalies

In S.J's method the time of Surf algorithm at best case is M^2 (3n +0.5). Now for each key point we need to have a comparator operator so that we can find our anomaly equal to 2M because for each Key Point In the first and second image we will have a comparator that does the job of validating whether or not that point is key.

$$M^2(3n+0.5) + \text{ANOMALY\_DETECTION\_TIME} = M^2(3n+0.5) + 2M \qquad (12)$$

Now suppose an anomaly value with a dispersion coefficient of 50% is reduced by the following equation:

$$M^2(3n+0.5) + 2M - \{0.5(M^2(3n+0.5))\} \qquad (13)$$

Assuming a final value of M = 1000 N = 64:

$$192500000+2000-0.5(192500000) = 96251000 = 96/2\ MI^2 \qquad (14)$$

In the best case scenario, the S.J.CHEN algorithm has actually improved by 50 percent, but it is not a real story. Because in the most images, the amount of recovered anomalies will be virtually lower. Also the anomalies removing may have negative effect on the matching quality. The cost of color classification and anomaly removal will be equal to the qualitative reduction of the comparison of key points.

3. Experimental Results

In this paper, two parameters are required to compare the performance of a comparison program:

• *Speed parameter quantization which is proportional to the runtime reduction.*

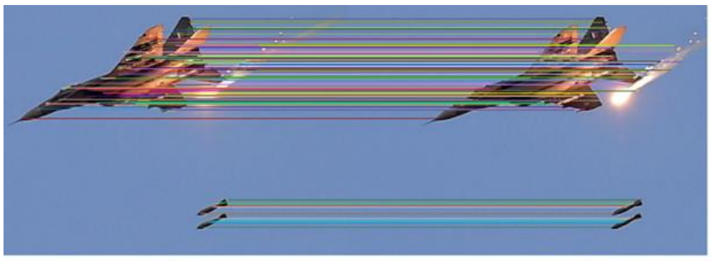
• *Preserving the quality parameter is the ability to accurately detect the program.*

In this qualitative and quantitative test, the following machine specifications were used for measurement:

• Intel Cpu 2.9 Ghz
• Memory 4.00Gb
• Matlab R2012b
• OS 8 64bit

Figure 10 shows the result of applying this algorithm to Figure 9 (a for SURF and b for proposed algorithm). We set the value 0.1556 for the amplitude in special minimization function. We know that in an ascending sorted array, finding the minimum is the best case for the least displacement. The execution time is shown in Figure 10 in the MATLAB profiler. Surf algorithm codes and the proposed algorithm are available from the following github:

https://github.com/esnkrimi/proposedSURF


A

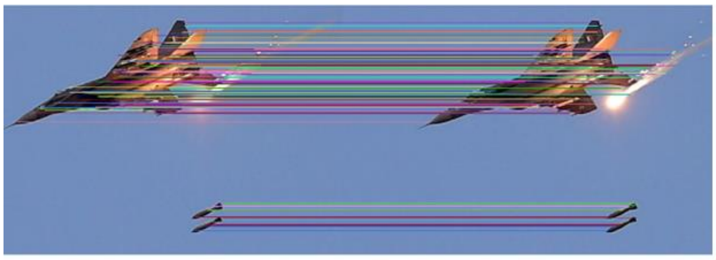
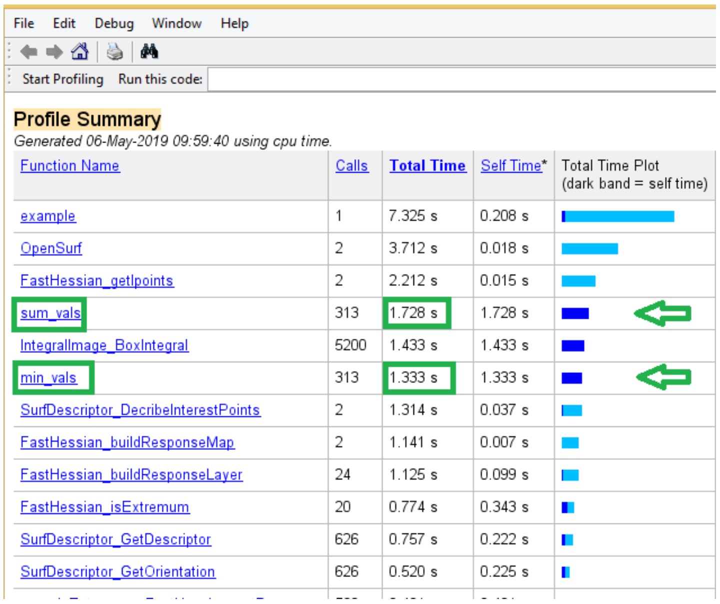

B

Figure 9: Results of implementation of the proposed algorithm and Surf



Profile Summary

Generated 06-May-2019 09:59:40 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
example	1	7.325 s	0.208 s	
OpenSurf	2	3.712 s	0.018 s	
FastHessian_getIpoints	2	2.212 s	0.015 s	
sum_vals	313	1.728 s	1.728 s	
IntegralImage_BoxIntegral	5200	1.433 s	1.433 s	
min_vals	313	1.333 s	1.333 s	
SurfDescriptor_DecribeInterestPoints	2	1.314 s	0.037 s	
FastHessian_buildResponseMap	2	1.141 s	0.007 s	
FastHessian_buildResponseLayer	24	1.125 s	0.099 s	
FastHessian_isExtremum	20	0.774 s	0.343 s	
SurfDescriptor_GetDescriptor	626	0.757 s	0.222 s	
SurfDescriptor_GetOrientation	626	0.520 s	0.225 s	

Figure 10: Matlab Analyzer Results for Proposed Algorithm and Surf

[۲] million instruction

4. Conclusion

In this paper we have made positive effects to the process of comparing the key points by some changes in matching step. Thus, instead of comparing each of the points by the sum of these points, we can use predictive concepts such as Min Max estimates. In this algorithm, we used the Hausdorff method and also designed a special minimization function, and we were able to achieve an acceptable value to improve its time complexity.

In the theory of the proposed algorithm, we found that the efficiency increase for the best case run time reduction was about 33% and the normal case 16% (Table 1).We decreased the run time to about 67% and 84% runtime of the Surf algorithm. We tested the theory by experimenting with the actual image and in MATLAB software. It was found that the runtime improvement reached 25%. That is, the runtime reached about 75% of the run time of the Surf algorithm, albeit due to the discrepancy between the theoretical and practical results. This competitive test, to the variables and conditions of application execution such as software vacuum such as operating system-level processes And Matlab debugger and software that incorporates techniques such as pagination or segmentation

Also, our algorithm performs worse than the S.J.CHEN algorithm, which was the last work to improve the Surf algorithm, on the other hand due to the domain independence of our algorithm and the S.J.CHEN algorithm as well as the non-interference of the input and output variables. Between our algorithm and S.J.CHEN, all the S.J.CHEN algorithm achievements can be exploited in our algorithm. Our algorithm outperforms Surf in our algorithm over S.J.CHEN in the worst case 1% and in the normal case 8% (Table 2).

References

Abdel Aziz Taha and Allan Hanbury 2015,"An Efficient Algorithm for Calculating the Exact Hausdorff Distance"

Aleš Hladnik ‹Helena Gabrijelčič Tomc (2016)" *SURF: Detection ‹description and matching of local features in 3d computer*"

Amin Mohamed Ashen ‹Dzulkifli Bin Mohammad (2013) "*Features Extraction for Object Detection Based on Interest Point*" 2716- 2722

Aomei Li ‹Wanli Jiang ‹Weihua Yuan ‹Dehui Dai ‹Siyu Zhang ‹ Zhe Wei)2017)" *An Improved FAST+SURF Fast Matching Algorithm*" 312-306

Darshana Mistry, Asim Banerjee (2017), "*Comparison of Feature Detection and Matching Approaches: SIFT and SURF*", ISSN: 2455-5703

David G. Lowe (2004) "Distinctive Image Features from Scale-Invariant Key points"

Edouard Oyallon ‹Julien Rabin (2015) "*An Analysis of the SURF Method*" 2105–1232

Ethan Rublee - Vincent Rabaud - Kurt Konolige - Gary Bradski (2011) "*ORB: an efficient alternative to SIFT or SURF*"

Hatem Mousselly Sergieh ‹Elod Egyed-Zsigmond ‹Mario Doller ‹David Coquil ‹Jean-Marie Pinon ‹and Harald Kosch ‹ INSA de Lyon (2017) *Improving SURF Image Matching Using Supervised Learning*"

Herbert Bay ‹Andreas Ess ‹Tinne Tuytelaars (2008) ‹ Luc Van Gool "*Speeded-Up Robust Features (SURF)"* 346–359

KADHIM H. KUBAN1 ‹WASAN M. JWAID (2005) "*A novel modification of surf algorithm for finger print*" 1992-8645

Karel Horak ‹Jan Klecka ‹Ondrej Bostik and Daniel Davidek (2017) "*Classification of SURF Image Features by Selected Machine Learning Algorithms*"

Maliheh Shabanzade ‹Morteza Zahedi and Seyyed Amin Aghvami(2011) " *Combination of local descriptors and global features for leaf*"

M. D. Wills (2007) "*Hausdorff Distance and Convex Sets* "UT 84408-1702

M. Hassaballah ‹Aly Amin Abdelmgeid and Hammam A. Alshazly (2016)" *Image Features Detection ‹Description and Matching*"

Pablo Fern´andez Alcantarilla ‹Adrien Bartoli ‹and Andrew J. Davison (2012) "*KAZE Features* "

Parmar Roshan R, Dr Gopal R,Kulkarni ,Lokesh Gagani (2015)"*Improving accuracy in image registration with sift and ransac*" 2349-6002

Peter Abeles (2012)" *Speeding Up SURF*" 2349-6002

Philippe Dreuw ‹Pascal Steingrube ‹Harald Hanselmann ‹ Hermann Ney (2007)" *SURF-Face: Face Recognition under Viewpoint Consistency Constraints*"

P M Panchal ‹S R Panchal ‹S K Shah (2013) "*A Comparison of SIFT and SURF*": 2320 – 9798

Seo Li Kang ‹Woo Kyung Lee (2014)" *SAR Image Change Detection Using SURF Algorithm*"

S. J. Chen ‹S. Z. Zheng ‹Z. G. Xu ‹C. C. Guo ‹X. L. Ma (2018) "*An improved image matching method based on surf*"

Tarun Tyagi ‹Vishal Mishra (2016) "2D Gaussian Filter for Image Processing: A Study" ISSN: 2349-784x

Upendra Singh ‹Sidhant Shekhar Singh ‹Manish Kumar Srivastava (2015)"Object Detection and Localization Using SURF Supported By K-NN"