Investigating SIFT Algorithm Performance on Image Deformations

Jaculine Medley
Toronto, ON
jmedley@ryerson.ca

December 18, 2021

Abstract— This project analyzes the performance of the SIFT (Scale Invariant Feature Transform) algorithm on image deformations. Anand Vijay's MATLAB implementation of David Lowe's SIFT algorithm was used.[7] A copy of the code can be found here https://github.com/CodeNameJacks/SIFT-Algorithm. The transformations performed on a single image were changes to intensity, scale, and rotation. The algorithm was applied to the image before and after image transformations and the extracted features were compared to determine how accurately the algorithm detected the image after the image transformation.

INTRODUCTION

The ability to match objects in an image or visual field is a fundamental aspect of computer vision. Image matching an essential requirement when trying to solve many real-world computer vision problems such as object recognitions, image retrieval, robot localization and building panoramas, to name a few.[1] Image matching algorithms can be global feature-based or local feature-based, with local feature-based being more stable.[1] In 2004, David Lowe (Lowe) proposed a Scale Invariant Feature Transform (SIFT), a local feature based algorithm based on detecting local key points and matching descriptions of said points. Key points are first extracted and transformed to a feature vector. A new image's key points are also extracted and transformed to a feature vector and matched to the first image by matching the descriptors in the feature vectors based on the Euclidean distance between the feature vectors. SIFT features that are extracted are highly distinctive and invariant to changes in scale, rotation, illumination.[2] There are many modifications to SIFT as well as similar algorithms such as SURF (Speeded Up Robustness Features), BRIEF (Binary Robust Independent Elementary Features), ORB (Oriented Fast and Rotated BRIEF), which were designed for specific applications and to compute faster the original approach.[3] While there are other feature detection algorithms such as Harris Corner or HOG, SIFT provides and advantage in that it is not affected by the size or the orientation of the image. [4]

In this paper, the matching performance of SIFT algorithm applied to different image transformations is examined and evaluated. We conduct experiments by making scale, rotation and illumination changes to an image and evaluate the SIFT algorithms performance by the accuracy of the number of matching feature points that were extracted between the original and the transformed image. We will be using Anand Vijay's MATLAB implementation of Lowe's algorithm.

TECHNICAL PART

The SIFT algorithm transforms an image into a feature vector in four steps.

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

Scale-space extrema detection

Feature locations are determined as the local extrema of Difference of Gaussians. Keypoints are detected through a cascading scale filtering approach where each cascade is called an octave. "For each octave of scale, the original image is repeatedly convolved with Gaussians to create a set of scale space images...Adjacent images are subtracted to produce difference of Gaussian images".[6] Each Gaussian image is down-sampled by 2 and the process is then repeated. The difference of Gaussian is the difference of two nearby scales separated by a constant factor k. [6]

$$\begin{aligned} D(x,y,\sigma) &= \big(G(x,y,k\sigma) - G(x,y,\sigma)\big) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad [6]$$

where * is the convolution operation in x and y, and

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad [6]$$

Keypoint Localization

Here we are trying to find the key points form the image that can be used for feature matching. In essence, we are finding the local maxima and local minima for the image.[4] This part is done in two steps. The first step is to find the local minima and maxima. This is done by comparing each pixel with its 26 neighbours and determining whether that pixel is a local minima or maxima.[4] That determination is made based on whether the pixel value is the highest or lowest in its neighbourhood. Keypoints are then selected by removing low contrast keypoints or keypoints that lie close to the edge.[4] This is done by rejecting pixels whose values fall below a threshold value.

Orientation Assignment

To establish invariant to rotation, a consistent orientation is assigned to each keypoint. A Gaussian smoothed image is selected using the scale of the keypoint. For each image at that scale the gradient magnitude and orientation is precomputed using pixel gradient in the x and y directions.[4] [6]

$$m(x,y) = \sqrt{(Gx)^2 + (Gy)^2} \qquad [4]$$

$$\theta(x,y) = tan^{-1}(Gy/Gx) \qquad [4]$$

Magnitude is the intensity of the pixel and orientation is the direction.[4]

An orientation histogram is then formed where the highest peaks are detected along with local peaks within 80% of the highest peak to create a keypoint with that orientation.[6]

Keypoint Descriptor

"A descriptor is formed from a vector containing the values of all the orientation histogram entries".[6] This is called the feature vector. Modifications are then made to the feature vector to make it invariant to changes in brightness. Those modifications consist of normalizing the feature vector to unit length followed by the filtering out any values greater than 0.2 and then renormalizing to unit length.[6]. This process results in invariance to constant changes in brightness as well as non-linear changes in illumination.

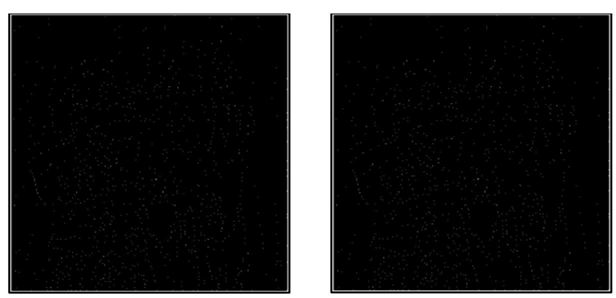
As a result of the four steps, the image pixels become invariant to scale, rotation, and illumination. The result is a feature vector or a set of image descriptors that are stable and can be reliably used for image matching. The extent to which the feature vector when compared to the feature vector of a new image matches, determined the degree to which an object or image can be recognized.

EXPERIMENTS

Performance of SIFT algorithm for correct matches was evaluated. The image in Figure 1 was used and turned into a grayscale image.

Figure 1: Original image and its greyscale equivalent

The SIFT algorithm was applied to the greyscale image to extract its feature descriptors. The descriptors were collected and placed in an array. Then various scale, rotation and illumination transformations were performed on the greyscale to produce a new image. The SIFT algorithm was applied to the image and its feature descriptors were also extracted and were placed in a different array. Each element in the first array were compared with all the elements in the second in the second array. A counter variable kept count of the matching elements and would increment by one each time there was a match. The total number of matches was divided by the total number of features descriptors to arrive at an accuracy percentage for the image matching. Figure 2 shows an image of the keypoints identified in the original and an image transformed for scale. In all transformation cases, the images of the identified keypoints looked the same. As such, no further image showing the identification of keypoints will be shown.



Original image | Scale Transformed image at σ=1

Figure 2: Keypoints Identified in Original image and Scale Transformed Image at σ = 1

Results of Scale Transformations

Scale transformations were performed by changing the value of σ while keeping all other factors constant. A σ=1.6 resulted in a 100% accuracy in the image match and was the base value used in the algorithm. σ was increased and decreased in increments of 0.3 and the accuracy percentage was recorded. Figure 3 shows the new images resulting from the transformation of the original greyscale image at various values of σ. Table 1 contains the percent accuracy of the matching feature descriptors between the original image and the scale transformed image for various values of σ.

Original greyscale image, σ=1.6

Transformed image, σ=1.3

Transformed image, σ=1.0

Transformed image, σ=1.9

Transformed image, σ=2.2

Figure 3: Original Image and Scale Transformed Images at Various Values of σ

σ value	1	1.3	1.6	1.9	2.2
matching accuracy %	61%	86.62%	100%	90.64%	86%

Table 1: Percent Accuracy of Matching Feature Descriptors Between the Original Image and the Scale Transformed Image at Various Values of σ.

The results show that a slight reduction in accuracy in the number of match feature descriptors as on moves away in either direction from the baseline σ value of 1.6. Accuracy declined more rapidly when reducing the value of σ than it did when increasing the value of σ. In both an increase and decrease in scale, the SIFT algorithm demonstrated a partial invariance to scale.

Results of Rotation Transformation

Rotation transformations were performed by changing the value of the angle of rotation while keeping all other factors constant. Initially, small rotations were made above and below the angle 0º. Then experiments were conducted at rotations a multiple of 22.5 degrees. The increment of 22.5 degrees was chosen because according to Karami et al., " SIFT algorithm performs the best at rotations equal to multiple integers of 22.5 degrees which the size of each bin in the histogram of gradient employed by the SIFT algorithm".[5]. Figure 4 shows the new images resulting from the transformation of the original greyscale image at various degrees of rotation. Table 2 contains the percent accuracy of the matching feature descriptors between the original image and the scale transformed image for various values of σ.

Original greyscale image, no rotation

Transformed image at -5 degrees rotation

Transformed image at 5 degrees rotation

Transformed image at 7 degrees rotation

Transformed image at 10 degrees rotation

Transformed image at 15 degrees rotation

Transformed image at 22.5 degrees rotation

Transformed image at 45 degrees rotation

Figure 4: Original Image and the Rotation Transformed Images at Various Degrees of Rotation

degrees rotation	-10	-5	5	7	10
matching accuracy %	18.51 %	83.02 %	81.48 %	73.46 %	17.34 %

degrees rotation	22.5	45	60	80
matching accuracy %	52.26 %	27.16 %	20.98 %	18.52 %

Table 2: Percent Accuracy of Matching Feature Descriptors Between the Original Image and the Rotation Transformed Images at Various Degrees of Rotation

The results from this experiment show that for small rotations, those below +/- 5 degrees, the SIFT algorithm was partially invariant to rotations. For rotations greater than +/- 10 degrees, the SIFT algorithm performed poorly. Also, in this experiment the SIFT algorithm did not perform best at increments of 22.5 degrees as Karami et al.[4] had observed in their experiments.

Results of Illumination Transformations

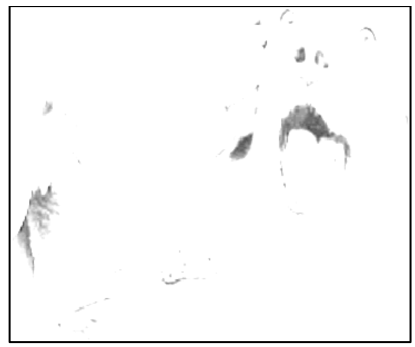
Rotation transformations were performed by multiplying the image intensity value at increments of 5 while keeping all other factors constant. Figure 5 shows the new images resulting from the transformation of the original greyscale image at various values multiples of 5. Table 3 contains the percent accuracy of the matching feature descriptors between the original image and the illumination transformed image for various values of σ.

Original greyscale image, no illumination change

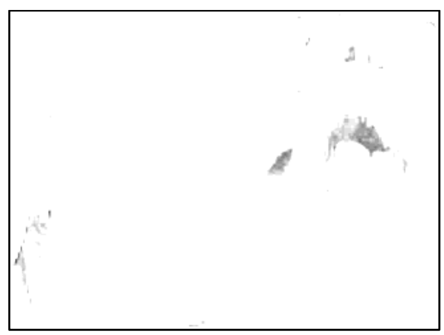
Transformed image at intensity multiplication factor = 1/5

Transformed image at intensity multiplication factor = 1/10

Transformed image at intensity multiplication factor = 5



Transformed image at intensity multiplication factor = 10



Transformed image at intensity multiplication factor = 15

Figure 5: Original Image and the Illumination Transformed Image at Various Multiplication Factors at Increments of 5.

illumination factor	1/5	1/10	5	10	15
matching accuracy %	99.89 %	99.89 %	99.79 %	99.79 %	99.79 %

Table 3 Percent Accuracy of Matching Feature Descriptors Between the Original Image and the Illumination Transformed Image at Various Multiplication Factors at Increments of 5.

The results from this experiment show the SIFT algorithm is invariant to changes in illumination.

Conclusion

In this experiment the performance of SIFT algorithms was evaluated a various scale, rotation, and illumination image deformations. Invariance to illumination was exhibited. However, the algorithm demonstrated partial invariance to low changes in rotation as well as scale and performed poorly when changes in rotation we greater than 10 degrees in the both the positive and negative directions. Perhaps the implementation of the algorithm used played a factor and that better results for scale and rotation transformation would have been observed with a different implementation.

References

[1] Wu, Jian et al. (2013) A Comparative Study of SIFT and its Variants. *Measurement Science Review,* Volume 13, No. 3. Retrieved December 5, 2021, from https://www.measurement.sk/2013/ Jian.pdf

[2] Alhwarin, F. et al. n.d. *VF-SIFT: Very Fast SIFT Feature* Matching Retrieved December 4, 2021, from https://link.springer.com/ content/pdf/10.1007%2F978-3-642-15986-2_23.pdf

[3] Hurtik, P. et al. n.d. *SIFT Limitations in Sub-image Searching*. Retrieved December 5, 2021, from https://www.researchgate.net/ publication/319328214_SIFT_Limitations_in_Sub-image_Searching

[4] Aishwarya, S. October 9, 2019. *A detailed Guide to the Powerful SIFT Technique for Image Matching (with Python code).* Retrieved December 8, 32021, from https://www.analyticsvidhya.com/ blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/

[5] Karami, E. et al. n.d *Image Identification Using SIFT Algorithm: Performance Analysis against Difference Image Deformations*. Retrieved December 4, 2021, from https://arxiv.org/pdf/ 1710.02728.pdf

[6] Lowe, D. January 5, 2004. *Distinctive Image Features from Scale-Invariant Keypoints.* Retrieved December 10, 2021, from https://people.eecs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

[7] Vijay, A. (2021). SIFT (Scale Invariant Feature Transform) (https://www.mathworks.com/matlabcentral/fileexchange/43724-sift-scale-invariant-feature-transform), MATLAB Central File Exchange. Retrieved December 4, 2021.

[8] Lowe. D., n.d. *Object Recognition form Local Scale-Invariant Features*. Retrieved December 4, 2021, from https://www.cs.ubc.ca /~lowe/papers/iccv99.pdf