Improving Image Feature Detection - The hybrid approach using SIFT

Ashwini Sharma
Graduate Student - MSAI
University of North Texas
Denton, Texas, USA
ashwinisharma@my.unt.edu

Venkata Kovvuri
Graduate Student - MSAI
University of North Texas
Denton, Texas, USA
venkatasrisatyakovvuri@my.unt.edu

***Abstract*—SIFT, a powerful feature extraction method for local image features, is packed with various strengths and advantages. It has been studied and enhanced in the past two decades resulting in numerous variants and different hybrid approaches available to use. These important variants which are already developed and implemented, having their unique strengths and weaknesses. Most important thing is to understand how they can be used to solve problems at hand. This paper explores not only the variants, but also the combination of different strong extractors with SIFT, which are also available to use. Outlines for creating a new descriptor have been proposed. A handy summarized list mentioning strengths and weaknesses of 14 useful extractors has been added for reference.**

***Index Terms*—SIFT, Feature Extraction, Hybrid Extractor, SIFT Variants, Image Features**

I. INTRODUCTION

Scale Invariant Feature Transform: SIFT [1] is a popular feature extraction technique that identifies and describes local features invariant to scale, rotation, and affine transformations. It detects key-points in an image and generates descriptors based on the local image gradients. SIFT was invented by David Lowe in 1999 and has been used effectively till present times.

This approach [1] transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. Lack of invariance to scale and sensitivity towards illumination change were real problems in previous approaches used for local feature generators.

Combination of scale invariance, robustness to transformations, efficient feature extraction, and verification process makes it a powerful and practical approach for object recognition tasks in real-world environments. SIFT identifies locations in image scale space that are invariant with respect to image translation, scaling, and rotation, and are minimally affected by noise and small distortions. The Gaussian kernel and its derivatives are being used as smoothing kernels for scale space analysis.

On the contrary, in this technique, few challenges which still need to be taken care are, sensitivity to noise, high memory requirement, intensive computation, limited rotation invariance, and non-uniform scale changes.

Moreover, with time, as technology in capturing images matured, the problem statements have changed drastically and there are scenarios where achieving high accuracy is simply not possible with naive SIFT. Large datasets, real time response, growing image sizes, incorporating color channel and texture in descriptor, requirement of enhanced rotational invariance are few of the examples.

A number of studies were carried out and different techniques were invented to enhance SIFT. These studies can be divided in 2 broad categories, one which works on SIFT variants, such as, ASIFT [2], PCA-SIFT [3], CSIFT [4], GSIFT [5] and the other one where hybrid methods were implemented which are combination of SIFT with some other technique like SURF [6] or ORB [7].

In this paper, we will discuss the various SIFT variants together with the advantages they bring, as discussed in section 2. We will be discussing a few of the hybrid methods such as, SIFT-ORB [8] and SIFT-FAST [9] in section 3. Section 4 discusses a general plan for how to proceed conceptually with a new hybrid approach, mentioning important parameters to consider while creating a custom combination of descriptors. We will have a comparison study [19] highlighting some of the descriptors which can be used to produce new combinations. Section 5 concludes this paper, followed by important references.

II. RELATED WORK

A. SIFT Variants

Numerous variants of SIFT [1] have been implemented so far such as ASIFT [2], PCA-SIFT [3], CSIFT [4], GSIFT [5], BSIFT [10] described in greater depths. These new techniques were developed to overcome earlier mentioned issues in SIFT. An example, BSIFT [10] is developed to improve computational efficiency by reducing the size of the descriptor which is handling 128 dimensional vector ($4 \times 4 = 16$ histograms each with 8 bins) by replacing gradient based descriptor with binary descriptor.

B-SIFT [10] is a binary version of SIFT, which uses binary descriptors instead of floating-point descriptors, resulting in reduced descriptor size and computational complexity, while maintaining or sometimes improving accuracy. This makes B-SIFT faster than SIFT, also it requires less storage. It comes with some disadvantages as well. B-SIFT is not as robust to noise as SIFT. It is more likely to be affected by small changes in the image, such as changes in lighting or camera position. it cannot represent the full range of features that SIFT can. There are fewer libraries and tools available for using B-SIFT, and it may be more difficult to find support if we have problems using it. B-SIFT is a good choice for applications where speed is important, but where robustness to noise is not a critical requirement.

Rotational invariance is another key parameter and in some of the studies, researchers focused on improving this feature. "Improving SIFT-based Descriptors Stability to Rotations" [11] is one of those approaches which improves SIFT by making it more robust to rotations.

This approach uses predefined discrete orientations, which can be easily derived by shifting the descriptor vector, instead of the original approach where it's needed to rotate the feature patch before computing the descriptor. This makes the descriptor less sensitive to changes in the orientation of the feature patch, which can be caused by factors such as changes in lighting or camera viewpoint. Descriptors used in this approach are called sGLOH and sGLOH+. This study [12] compared new descriptors with the SIFT descriptor on the Oxford image dataset. The proposed descriptors are more robust to rotations than SIFT, and they also achieve comparable results in terms of accuracy.

SIFT works on grayscale images and an important aspect of image carrying useful information is being missed, i.e. "Color Channel". Various studies tried to improve SIFT by incorporating color channels and **CSIFT** [4] is one of those. CSIFT [4] improves native SIFT by using color information to make the descriptors more robust to changes in lighting conditions. Proposed approach handles this by using a color invariant model to calculate the color descriptors for each interest point. This approach is based on the Kubelka-Munk theory, which models the reflected spectrum of colored bodies. This study is able to calculate the color invariants for a given interest point even if the lighting conditions are different. In simpler words, CSIFT is a more advanced version of SIFT that is able to identify features in images that are more resistant to changes in lighting conditions.

Other than mentioned areas, few studies focused on handling certain aspects of images like change in illumination and improving SIFT by reducing common key points. We will discuss these before moving to a hybrid approach. This method improves SIFT descriptor using insensitivity to changes in illumination. The **MGS-SIFT** [12] method uses a Power-Law Transform to create new samples of an image in different illumination conditions. These samples are then used to extract the SIFT features. The results show that the MGS-SIFT method has a considerable effect in increasing the number of matched points and it also increased the accuracy of classification.

The SIFT descriptor is improved in this method by using a normal color space and **subtractive clustering**. The normal color space helps to make the descriptor invariant against illumination variants, while subtractive clustering helps to reduce the number of key points, which can improve the efficiency of the algorithm.

There is another approach [13] that works on a similar illumination invariant approach, uses a normal color space and subtractive clustering. The normal color space helps to make the descriptor invariant against illumination variants, while subtractive clustering helps to reduce the number of key points. Subtractive clustering method converts the image to a normal color space, such as HSV or CIELAB. This helps to make the descriptor invariant against illumination variants, as these color spaces are less sensitive to changes in lighting. The image is then clustered into groups of similar pixels using a technique called subtractive clustering. This helps to reduce the number of key points, which results in improving the efficiency of the algorithm.

There are some experiments towards key point reduction and subtractive clustering is helpful in those approaches. One approach [14] improves SIFT outcome by reducing the number of key points that are similar to each other. This can be done by clustering the key points based on their features and then removing the points that are in the same cluster. This will result in a smaller set of key points that are more distinct from each other, resulting in improving the accuracy of SIFT. Working principle uses the following steps, calculate the features of each key point. These features can include the location of the key point, its orientation, and its scale. Cluster the key points based on their features. This can be done using a variety of clustering algorithms. Remove the key points that are in the same cluster. This will result in a smaller set of key points that are more distinct from each other. Disadvantage of this approach is increased execution time, as this involves a number of steps mentioned earlier, including calculating the features, clustering, and removal of the key points. These steps can be time-consuming, especially for large images. However, this approach improves the accuracy of SIFT, which can be beneficial if accuracy is a crucial parameter.

There are some other comparison studies also available to refer, such as "A Comparative Study of SIFT and its Variants" [15] which talks about SIFT, PCA-SIFT, GSIFT, ASIFT.

B. Hybrid SIFT (Combinations)

In the previous section we have mentioned variants of SIFT in order to minimize a few of the shortcomings. There is one more way other than making changes to SIFT, the hybrid approach, where we can also combine different techniques to generate more promising feature vectors. There are various techniques developed for different kinds of problems, few really useful and well known techniques are ORB [20], FAST [21], BRIEF [22], SURF [23] and LBP [24].

We will be discussing a few of the hybrid techniques highlighting the difference, pros and cons.

The first technique is a combination of **SIFT and ORB**. The proposed approach [8] uses both descriptors to improve the accuracy of image matching. SIFT and ORB are both effective descriptors with their own strengths and weaknesses. SIFT is more robust to changes in illumination and viewpoint and

ORB is faster than SIFT. The proposed approach achieves the benefits of both descriptors and can be more accurate, more robust, and more practical than individual SIFT and ORB. There are some drawbacks as well in terms of speed and memory usage. The proposed approach is slower and requires more memory than individual SIFT and ORB because it needs to compute and store the descriptors for both techniques. This approach can be preferred for applications where accuracy and robustness are more important than speed and memory usage. Some potential use cases are, aligning two or more images of the same scene so they can be compared or combined, analyzing medical images, such as X-rays, CT scans, and MRIs, to identify abnormalities, and understanding the content of a scene from an image or video.

The second technique is combining **LBP and SIFT**. This approach [16] first extracts both LBP and SIFT features from an image and then creates a two-dimensional feature table. Followed by sparse coding to encode each feature pair and combine them to create a final feature for the image. This hybrid technique is more robust to noise and occlusion, more discriminative and can capture more complex features. Also requires more computational power and is more difficult to interpret as we are combining two complex techniques.

The third method combines **SIFT and BRIEF**. This method [17] uses a post-filter to re-filter key points which are rejected in the first step. This post-filter uses BRIEF descriptor space and hamming distance to re-filter the key-points, which helps to increase the number of matched key-points without decreasing the accuracy. Hamming distance is a measure of the similarity between two BRIEF descriptors. The post-filter rejects key-points that have a high hamming distance to any of the key-points that have already been matched. The main advantages of this approach over the individual ones are, better handling of noise and occlusion, practical with large images, and can be used with both indoor and outdoor scenes while being faster at the same time. Certainly there are some disadvantages such as, more computational cost and more sensitivity to image scale changes.

Next, i.e. fourth in this list is a combination of **SIFT and FAST**. This approach [9] leverages the computational efficiency of FAST and robustness of SIFT and creates a hybrid technique to enhance the performance of feature extraction specifically for applications requiring indoor positioning. Instead of SIFT's gaussian pyramid for point detection, this method speeds up corner detection with FAST and quickly detects corners in image, which act as candidate feature points for next steps where SIFT is used to describe these points. SIFT calculates gradient orientations and forms 128 dimensional feature vectors for each feature point. The FAST-SIFT hybrid approach addresses the limitations of both individual algorithms and provides a faster and more efficient feature point extraction process by using FAST while benefiting from the accurate and robust feature descriptors provided by SIFT. The major advantages are improved accuracy, real-time performance of image feature extraction and enhanced stability. On the other hand, this approach introduces additional complexity to the implementation, requires deep understanding of the system, potentially miss out on some relevant feature points as we are using FAST for corner detection, limited rotational invariance as FAST lacks rotational invariance while SIFT has limited invariance and increased computational complexity in case FAST generates large number of feature points.

There are some other studies which also cover a similar hybrid approach for FAST and SIFT [25] and one which introduces CamShift with FAST and SIFT [26].

Last, i.e. fifth in this section is combining **SIFT with SURF**. This approach [18] discusses the use of hybrid algorithms specifically for aerial image stitching where this combination provides a more robust and efficient image stitching process. This combination improves accuracy, leads faster computation and handles large datasets compared to SIFT or SURF individually. When SIFT is used as detector and SURF as descriptor, it achieves better matching accuracy in this approach resulting in improved stitching quality. As SIFT is computationally expensive, especially while dealing with large datasets, SURF comes handy as a descriptor and reduces computational time. This approach uses brute-force matching with Euclidean distance, followed by the Lowe ratio test and RANSAC for homography estimation and image warping based on the matches, which provides a more reliable way to match features between images and creates seamless panorama. There are some disadvantages such as, increased complexity, sensitivity to content like low contrast and repetitive patterns or noise, and scale invariance.

III. Concept - Creating New Hybrid SIFT

As mentioned in the last section, there are multiple feature extraction techniques available with different strengths and weaknesses, working on different types of problem statements. We can create a new hybrid approach as per requirement by combining two or more techniques. It makes it easier if we know strengths and weaknesses of some of the widely used algorithms before moving to the outlines of combining them. Various studies have focused on comparing different feature extraction algorithms [27] and one of the study where the author compared 14 algorithms such as SIFT, SURF, ORB, BRISK, KAZE, AKAZE and more for matching extremely variant image pairs [19] has been referred, where "Table I" summarizes the information for future use.

A. Concept

As we have seen in the last sections, a number of notable works are already done for hybrid descriptors. Main important concept to understand is, we can make hybrid descriptors by combining any of the available algorithms. However, the combination mainly depends on the end goal of application.

We can create a hybrid descriptor by combining multiple feature descriptors. For example by combining SIFT, SURF, FAST, and ORB we can create a new hybrid descriptor. The idea behind creating a hybrid descriptor is to leverage the complementary strengths of each individual descriptor to achieve a more robust and discriminative representation for

TABLE I
COMPARING FEATURE EXTRACTION ALGORITHM

Algorithm	***Strengths***	***Weaknesses***
SIFT	Very robust to scale, rotation, and illumination changes	Slow and computationally expensive
SURF	Faster than SIFT	less robust to noise and blur
ORB	Very fast and computationally efficient	Not as robust to scale, rotation, and illumination changes as SIFT, SURF, KAZE, and AKAZE
BRISK	Very fast and computationally efficient, and more robust to noise and blur than ORB	Not as robust to scale, rotation, and illumination changes as SIFT, SURF, KAZE, and AKAZE
FAST	Very fast and computationally efficient	but not very robust to noise and blur
KAZE	Faster than SURF and SIFT, and more robust to noise and blur	Not as robust to scale and rotation changes as SIFT
AKAZE	Faster than KAZE, and more robust to scale and rotation changes	Not as robust to noise and blur as KAZE
AGAST	Fast and computationally efficient	Not as robust to scale, rotation, and illumination changes as SIFT, SURF, KAZE, AKAZE, ORB, and BRISK
MSER	Robust to noise and blur	Not very fast or computationally efficient
MSD	Very robust to noise and blur	Not very fast or computationally efficient
GFTT	Fast and computationally efficient	Not very robust to noise and blur
Harris Corner Detector based GFTT	Fast and computationally efficient and more robust to noise and blur than GFTT	Not as robust to scale, rotation, and illumination changes as SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, FAST, MSER, and MSD
Harris Laplace Detector	Very robust to scale, rotation, and illumination changes	Slow and computationally expensive
CenSurE	Very robust to scale, rotation, illumination changes, and noise and blur	Slow and computationally expensive

various computer vision tasks. This hybrid version will have the combined strengths, however, there will be disadvantages such as complexity and computation.

Here's a general outline of how we could create a hybrid descriptor using SIFT, SURF, FAST, and ORB:

- **Feature Detection:** Apply SIFT, SURF, and FAST to detect keypoints in the image as each one has its strengths in handling different types of image structures and patterns. At the end, we will have three sets of keypoints, one for each detector.
- **Feature Description:** For each set of keypoints (SIFT, SURF, and FAST) from the last step, we compute the respective descriptors. For example, SIFT will generate a 128-dimensional gradient-based descriptor, SURF will create a 64-dimensional descriptor, and FAST keypoints will have no descriptors by default. However, we can use another descriptor like ORB to describe FAST keypoints. This is all we have seen in section 3.
- **Combining Description/Features:** Concatenate the descriptors from the last step from each method into a single, long feature vector. The dimension of this hybrid descriptor will be the sum of dimensions of all individual descriptors.
- **Feature Matching and Recognition:** Finally, this hybrid descriptor can be used for tasks such as image matching, object recognition, or any other application that requires feature representations.

It's important to note that the success of a hybrid descriptor depends on the specific task and the characteristics of the dataset. Combining multiple descriptors can potentially lead to a more powerful representation, it may also introduce some redundancy or noise. Thus, it's essential to thoroughly evaluate the performance of the hybrid descriptor on the given specific problem.

Additionally, we can experiment with different combinations and feature selection techniques to determine which combination of descriptors works best for the problem statement. Feature selection or dimensionality reduction techniques, such as PCA (Principal Component Analysis) or LDA (Linear Discriminant Analysis), could be employed to further optimize the hybrid descriptor.

IV. Conclusion

In this paper, we studied SIFT as a feature extraction technique highlighting its different variants and combined hybrid approaches with other descriptors. Depending on the problem statement, a feature extraction algorithm can be picked which is based on SIFT. For example, if speed and descriptor size is a challenge and robustness to noise is not on priority, the BSIFT variant can be picked for extraction. Similarly, real time detection with robustness of SIFT is the requirement, SIFT - FAST hybrid approach is the answer. This paper helps to identify the correct method to pick for feature extraction and also proposes an outline in order to develop a new hybrid feature extraction algorithm. A Summary explanation from different studies and a comparison table between more than ten feature extraction algorithms has been compiled in this paper.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2.

[2] Morel, J.M., Yu, G. (2009). "ASIFT: A new framework for fully affine invariant image comparison," SIAM Journal on Imaging Sciences, 2 (2), 438-469.

[3] Yan Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Washington, DC, USA, 2004, pp. II-II.

[4] Abdel-Hakim, Alaa and Farag, Aly. (2006). "CSIFT: A SIFT descriptor with color invariant characteristics," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2. 1978 - 1983. 10.1109/CVPR.2006.95.

[5] E. N. Mortensen, Hongli Deng and L. Shapiro, "A SIFT descriptor with global context," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 184-190 vol. 1, doi: 10.1109/CVPR.2005.45.

[6] Bay, H., Tuytelaars, T., Van Gool, L. (2006). “SURF: Speeded Up Robust Features,” In: Leonardis, A., Bischof, H., Pinz, A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg.
[7] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.
[8] E. G. Andrianova and L. A. Demidova, “An Approach to Image Matching Based on SIFT and ORB Algorithms,” 2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation, 2021, pp. 534-539, doi: 10.1109/SUMMA53307.2021.9632214.
[9] X. Fu, K. Zhang, C. Shen, J. Zhu and L. Zang, “Recognition and Location Based on Fusion of FAST Algorithm and SIFT Algorithm,” 2021 IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 2021, pp. 422-426.
[10] Z. -S. Ni, “B-SIFT: A Binary SIFT Based Local Image Feature Descriptor,” 2012 Fourth International Conference on Digital Home, Guangzhou, China, 2012, pp. 117-121.
[11] F. Bellavia, D. Tegolo and E. Trucco, “Improving SIFT-based Descriptors Stability to Rotations,” 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 3460-3463.
[12] Javanmard Alitappeh, Reza and Mahmoudi, Fariborz. (2013). “MGS-SIFT: A New Illumination Invariant Feature Basedon SIFT Descriptor,” International Journal of Computer Theory and Engineering. 99-103.
[13] Javanmard Alitappeh, Reza and Jeddi, Kossar and Mahmoudi, Fariborz. (2012). “A New Illumination Invariant Feature Based on SIFT Descriptor in Color Space,” Procedia Engineering. 41. 305–311.
[14] Javanmard Alitappeh, Reza and Jeddi, Kossar and Mahmoudi, Fariborz. (2012). “Key point reduction in SIFT descriptor used by subtractive clustering,” 2012 11th International Conference on Information Science, Signal Processing and their Applications, ISSPA 2012. 906-911.
[15] Wu, Jian, Cui, Zhiming, Sheng, Victor S., Zhao, Pengpeng, Su, Dongliang and Gong, Shengrong. “A Comparative Study of SIFT and its Variants,” Measurement Science Review, vol.13, no.3, 3913, pp.122-131.
[16] S. Bai, “Sparse code LBP and SIFT features together for scene categorization,” 2014 International Conference on Audio, Language and Image Processing, Shanghai, China, 2014, pp. 200-205.
[17] V. Phuong Le and C. De Tran, “Key-point matching with post-filter using SIFT and BRIEF in logo spotting,” The 2015 IEEE RIVF International Conference on Computing and Communication Technologies - Research, Innovation, and Vision for Future (RIVF), Can Tho, Vietnam, 2015, pp. 89-93.
[18] R. M. Akhyar and H. Tjandrasa, “Image Stitching Development By Combining SIFT Detector And SURF Descriptor For Aerial View Images,” 2019 12th International Conference on Information and Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 209-214.
[19] S. A. Khan Tareen and R. H. Raza, “Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs,” 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2023, pp. 1-6.
[20] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571.
[21] Viswanathan, Deepak Geetha. “Features from accelerated segment test (fast),” Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK. 2009.
[22] Calonder, M., Lepetit, V., Strecha, C., Fua, P. (2010). “BRIEF: Binary Robust Independent Elementary Features,” In: Daniilidis, K., Maragos, P., Paragios, N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6314. Springer, Berlin, Heidelberg.
[23] Bay, H., Tuytelaars, T., Van Gool, L. (2006). “SURF: Speeded Up Robust Features,” In: Leonardis, A., Bischof, H., Pinz, A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg.
[24] T. Ojala, M. Pietikainen and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, July 2002.
[25] W. Lifang, G. Yuan and Z. Jingwen, “An Improved SIFT Algorithm Based on FAST Corner Detection,” 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 2013, pp. 202-205.
[26] X. Xiao, J. Wang, Q. Shen and Y. Wang, “An Improved CamShift Algorithm Based on FAST-SIFT Feature Detection Matching,” 2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP), Singapore, 2018, pp. 64-68.
[27] H. -J. Chien, C. -C. Chuang, C. -Y. Chen and R. Klette, “When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry,” 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, New Zealand, 2016, pp. 1-6.