FISB: Feature-based Image Stitching Benchmark

Abhishek Rajora
Indian Institute of Technology, Jodhpur
rajora.1@iitj.ac.in

Abu Shahid
Indian Institute of Technology, Jodhpur
shahid.3@iitj.ac.in

Abstract

The image stitching technology is broad and active research area in image processing and computer vision. This paper focuses on breaking the thresholds set by some of the challenging problems in this area. This paper presents a detail study of feature based image stitching algorithms with different evaluation metrics. We are in process of developing an image stitching pipeline for creating panoramic images from multiple input images. The proposed method involves several stages, including feature extraction, matching, and blending. For evaluation, Google Landmarks database was used. In addition, a custom data set is made with images taken from different viewpoints and varying illumination. This data set is used to evaluate the effectiveness of the proposed pipeline and to fine-tune the pipeline's parameters. The performance of the proposed pipeline is evaluated using both objective and subjective measures, including accuracy, speed, and visual quality. Experimental results show that the proposed pipeline can effectively stitch images and produce seamless panoramas. The pipeline is scalable and can be applied to a wide range of applications, such as surveillance, virtual reality, and cartography.

1. Introduction

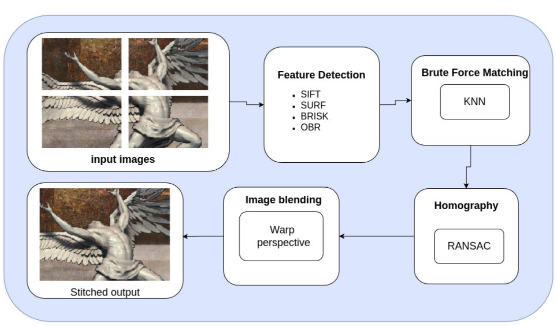

Figure 1. FISB pipeline

One of the major limitations of current image stitching algorithms is their inability to handle images taken from different perspectives, with different illuminations, or when images are rotated. Additionally, many of the basic and simple implementations of these algorithms can only stitch images in one orientation, such as horizontal or vertical. This restricts their usefulness for more complex applications, such as creating panoramas or stitching images from multiple viewpoints.

In this paper, we propose a pipeline for image stitching that incorporates several popular feature-matching algorithms, including SIFT, SURF, BRISK, and ORB. We employ brute force matching with K-nearest neighbor (KNN) algorithm to remove false positives. Following feature matching, we utilize homography to combine overlapping features of images using the RANSAC algorithm. The pipeline determines whether the images are to be stitched based on a probabilistic overlap threshold. The effectiveness of our proposed pipeline is demonstrated through experiments on a variety of images.

1.1. Contributions

I. Abhishek Rajora:

i) Implementing SURF and ORB
ii) Bruteforce matching with KNN
iii) Image Blending

II. Abu Shahid:

i) Implementing SIFT and BRISK
ii) Implementing Homographic Transformation with RANSAC
iii) Dataset creation and benchmarking

2. Related Work

In image stitching algorithms, the choice of feature extraction algorithm is critical for accurate and efficient image alignment.

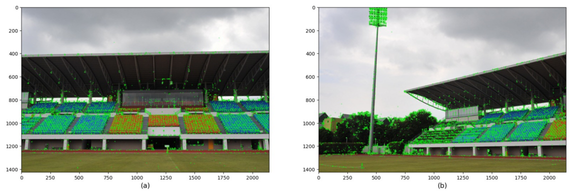

Figure 2. Feature Detection using SIFT

Figure 3. Brute Force feature matching with KNN

2.1. Feature Detection Algorithm

2.1.1 SIFT

SIFT (Scale-Invariant Feature Transformation) is a widely used feature extraction algorithm that detects and describes scale-invariant keypoints in images. SIFT detects and describes scale-invariant keypoints in images by identifying and describing distinctive local image features. It is based on the idea that local features that are invariant to scale, orientation, and illumination are more robust to image changes and more informative for matching across different images.

2.1.2 SURF

SURF (Speeded Up Robust Features) is an extension of SIFT that uses a faster Hessian-based approach for keypoint detection and orientation estimation, making it more efficient than SIFT. SURF works by computing the local intensity of an image in different directions and scales, then identifying and describing keypoints based on these local intensity patterns.

2.1.3 BRISK

BRISK (Binary Robust Invariant Scalable Keypoints) is a binary descriptor that uses a scale-space pyramid to extract keypoints and produces a compact binary descriptor that is efficient to compute and compare. BRISK uses a sampling pattern to determine the location and scale of keypoints, and a binary descriptor to describe the local image features.

2.1.4 ORB

ORB (Oriented FAST and Rotated BRIEF) is another binary descriptor that uses a combination of FAST keypoint detector and BRIEF (Binary Robust Independent Elementary Features) descriptor to extract and match keypoints. ORB uses a rotated patch to estimate the orientation of the keypoints, making it more robust to rotation than other feature extraction algorithms. It also uses a multi-scale pyramid approach for keypoint detection and descriptor computation.

2.2. Feature Matching

Brute force feature matching is a simple but effective technique for finding matching feature points between two images. The basic idea behind brute force matching is to compare each feature in one image to every feature in the other image and find the closest match based on some distance metric. One way to speed up the brute force matching process is to use a k-nearest neighbors (KNN) algorithm. KNN involves finding the k closest matches for each feature point in one image in the other image, based on the chosen distance metric. This can reduce the number of comparisons needed and improve the speed of the matching process. This integration of KNN allows us to eliminate false positives identified in brute force matching between image features.

2.3. Geometric Transformation

We utilise the matched keypoints to estimate the homography matrix that aligns the images. Homography estimation is a crucial step in image stitching, as it determines the transformation required to warp the images into a common coordinate system. Given the matched feature points, the geometric transformation between the two images is estimated using a robust estimation algorithm such as Random Sample Consensus (RANSAC). The transformation can be a homography or affine transformation, depending on the number of matching points and their distribution.

2.4. Image Blending

Figure 4. Image Blending with warp perspective

Once the homography matrix is estimated, we can use it to warp one of the images onto the plane of the other image, so that the two images are aligned. However, the resulting image may contain visible seams or artifacts due to

differences in brightness, color, or texture between the two images.
To address this issue, we can use image blending techniques to smooth out the seams and create a more seamless transition between the images. One common approach is to use a weighted average of the pixel values in the overlapping region, where the weights depend on the distance of each pixel from the boundary of the overlapping region.
Finally, Image blending with warp perspective is utilised in the pipeline, which allows us to create seamless panoramas from multiple overlapping images. It requires accurate estimation of the homography matrix and careful selection of the blending technique to ensure a smooth and seamless transition between the images.

3. Progress

To evaluate the performance of each method, we conducted experiments using the Google Landmark dataset. The Google Landmark dataset however have will have sub-images in a single perspective and have global illumination. To test the robustness of the proposed method, we also created our own dataset (currently have 37 scenes). Each set is made up of multiple sub-images having varying illumination, orientation, and perspective and a super image (to be used for validation). We measured each method's accuracy, computational complexity, and robustness regarding the number of correctly matched feature points, the time required for feature extraction and matching, and the sensitivity to image noise and illumination changes. However, to show the mean trends, the exercise needs to be carried with more scene images with the end-to-end pipeline.

4. Conclusion

Overall, the choice of feature extraction algorithm depends on the specific application requirements, such as the desired accuracy and computational efficiency. In scenarios where accuracy is the top priority, SIFT or SURF may be more suitable, while in real-time applications, BRISK or ORB may be preferred due to their efficiency. The pipeline so far is not end-to-end. We plan to reinforce the pipeline and further modularize it for ease of use. implementing more feature detection algorithms such as MSER is planned. We also target to incorporate global homography in our proposed pipeline. We plan to explore further blending and wrapping techniques. And finally test and benchmark our pipeline with already available implementations such as AutoStitcher and those offered by OpenCV, using evaluation metrics such as Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSI), Feature-based Similarity Index (FSI), and Signal to Reconstruction Error ratio against our custom and publically available datasets. We also plan to expand our in-house dataset to better present the benchmarks.