International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
OPEN ACCESS
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST

Generating Panoramic Image by Image Stitching Based on ORB Feature Detection Technique

A. Krishna Mohan [1], P. Sasirekha[2], K. Lavanya[3], N. Jithin Manjunadh[4], M.Venkata Manoj [5], N. Ajay[6]
[1]Assistant Professor, Department of Electronics and Communication Engineering, SV College of Engineering (SVCE), Tirupati, A.P. India
[2,3,4,5] UG Students, Department of Electronics and Communication Engineering, SV College of Engineering (SVCE), Tirupati, A.P. India

ARTICLEINFO

Article History:

Accepted: 20 Feb 2024
Published: 03 March 2024

Publication Issue :
Volume 11, Issue 2
March-April-2024
Page Number :
59-66

ABSTRACT

Generating panoramic images through image stitching is a crucial task in computer vision, enabling the creation of immersive and wide-field views from a series of overlapping images. This project focuses on implementing an image stitching algorithm based on the ORB (Oriented FAST and Rotated BRIEF) feature detection technique to seamlessly merge multiple images into a panoramic view. The proposed approach leverages the efficiency and robustness of ORB feature detection to identify keypoints in each input image. These keypoints serve as distinctive landmarks that facilitate accurate matching between overlapping regions of adjacent images. By employing geometric transformation methods and robust matching algorithms, the detected keypoints are used to estimate the homography transformation between pairs of images. This homography transformation aligns the images into a common coordinate system, enabling seamless blending and integration of overlapping regions. The implementation is performed using the OpenCV library in Python, providing a flexible and accessible platform for developing the image stitching algorithm. Experimental results demonstrate the effectiveness of the proposed approach in generating high-quality panoramic images with minimal computational overhead. Overall, this project presents a novel image stitching solution based on the ORB feature detection technique, offering a robust and efficient method for generating panoramic images. The integration of ORB feature detection with geometric transformation and blending techniques contributes to the advancement of image stitching algorithms, opening up possibilities for various applications in fields such as photography, virtual reality, and autonomous navigation.

Keywords : Image Stitching, Open CV, Python

I. INTRODUCTION

Image stitching or photo stitching is the process of combining multiple photographic images or frames with overlapping area of view to produce a photomosaic high- resolution image performed with help of computer software. To create seamless high resolution image, requires overlapping field between two images and identical exposures. Image stitching enables the combination of multiple shots to create a larger picture that is beyond the normal aspect ratio and resolution of the camera's individual shots. The most popular use of image stitching is in the creation of panoramic images, often used for landscapes. Wide-angle and super-resolution images created by image stitching are used in artistic photography, medical imaging, high-resolution photo mosaics, satellite photography and more. Image stitching is widely used in camcorders, digital maps, satellite photos, medical imaging, video stitching applications. Major issues in image stitching are presence of parallax, seamless image under varying illuminating conditions, scene motion, and exposure differences. Human visual system has a field of view of around 135 x 200 degrees where as a typical camera has a field of view of only 35 x 50 degrees. Therefore, to create a panoramic image, multiple camera images are stitched together and provides larger field of view[2]. The quality of image stitching is calculated by the resemblance of the overlapped portion of the stitched image and the visibility of the seam between the stitched images [3]. Image stitching is the challenging field in image processing field. It can be used widely in real time applications as satellite imaging, video stitching, advanced driver assistance

The organizational framework of this study divides the research work in the different sections. The Literature survey is presented in section 2. In section 3 and 4 discussed about Existing and proposed system methodologies. Further, in section 5 shown Simulation Results is discussed and Conclusion and future work are presented by last sections 6.

II. LITERATURE SURVEY

In the last many years, there are many researchers proposed algorithms, techniques, technologies for image stitching system. For example,

Levin and Weiss [29] proposed several cost functions to improve the. quality of stitching. Their approach is validated in the various applications, such as generation of panoramic images, in object blending, and for the removal of compression artifacts. The goal of a image stitching algorithm is to generate a visually acceptable panorama with two required properties. First, the stitched image and the input images should be as similar as possible, geometrically and photometrically. Second, the image produced should be seamless. Authors proposed several algorithms with numerous cost functions to meet these requirements. To create attractive panorama the seam region should be invisible. The quality of stitching in the seam region is measured in the gradient domain.

Brown and Lowe [32] utilizes the SIFT algorithm for a feature-based image stitching system. In the first step, SIFT features are extracted and matched between the images using panoramic recognition algorithm and then it finds overlapping or matching elements. It is necessary to match each image to a small number of neighboring images .It can offer a better solution for the image geometry. Then, a set of inliers is selected using RANSAC that are matched with aHomography between theimages. The goal of the Color correction method is to transfer color properties from source to target image. Fabio Bellavia and Carlo Colombo Proposed a new framework for classifying the color correction[25].

Cheng-Ming,Shu-Wei, jyun-Hong introduced an efficient image stitching algorithm. It provides better image quality and requires less computational time for real time stitching for a continuous image sequences. For this they utilized dominant frames instead of all.

Michael DuckjuneKim and Jun Ueda proposed[26] a rapidly moving robotic vision system to generate fast panoramic image. This vision system is used to receive blurry images and then recovered by real time de blurring method. These images are stitched together to create a panorama image using dynamic basedapproach .This is applicable when the images are received at the rate of 30 frames per second.

Chuan Li a, Zhi-YongLiu,,n, Xu Yang , Hong Qiao , Jian - Hua Su proposed an algorithm to improve the quality of noisy or contaminated images[31].

For concrete bridge inspection[33] , the current techniques are time consuming and dangerous. RenpingXie, Jian Yao*, Kang Liu, Xiao huLu, Yahui Liu, Menghan Xia, QifeiZeng proposed a novel stitching method which combines 2 D and 3 D image location point features to reducedrift.

III. IMAGE STITCHING

Image stitching basically divided into three main components as calibration, image registration, and blending [8] asshown in Fig. 1. External and internal camera parameters are estimated in camera calibration. Multiple images are compared in image registration and it find out the translations to align the images. After registration, in Blending, multiple images are merge together to create a single, large image In Brief discussion of these three components is given below. Location and orientation of the ideal reference frame and camera reference frame are the extrinsic parameters and pixel coordinates of the defined window of an image and the corresponding coordinates in the camera reference frame [9] can be linked in intrinsic parameters.

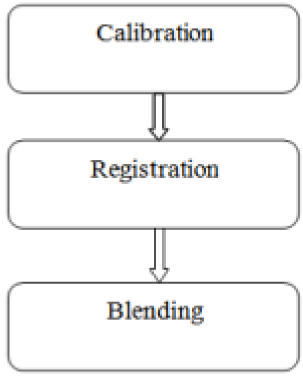

Figure 1: The main components of image stitching

1.Calibration

Exposure differences and optical defects results the differences between ideal lens model and camera lens combination. Image calibration minimizes this difference[8]. 3-D image can be reconstructed from the pixel coordinates of the image by recovering the intrinsic and extrinsic parameters of a camera.

B. Registration

Image registration is the core of an image stitching process. It is the process of transforming different sets of data as multiple photographs, sensor images etc. It creates geometric correspondence between the images and hence we can compare these images and can apply other steps properly. [20]. In this technique, two or more images which are captured from different point of perspectives are aligned.

C. Blending Image

Blending executes the adjustment figured out in image calibration as remapping of the images, color correction between images. Images are merged to create a seamless large image. The seam while stitching can be reduced with gain adjustment. It minimizes the difference between the overlapping pixels intensities of the two images. Alpha feathering and Gaussian pyramid are the two well knownmethods of blending[10]. The cases where well aligned image pixels are present with only differencein intensity between two images, then alpha blending works extremely well. Gaussian pyramid [20] is another popular approach where images at two different frequency bands are merged together and then filtered together.

IV. PROPOSED METHOD

Image stitching based on input images involves several key steps, including feature extraction, feature matching, homography estimation, image warping, and finally, the creation of a panoramic image.

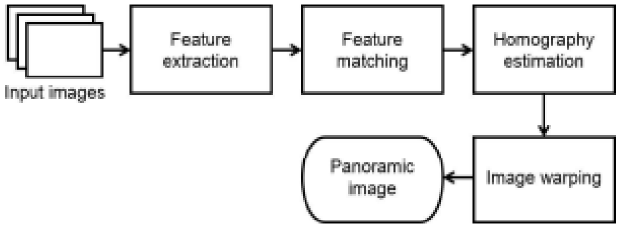

Figure 2. feature based image stiching

1. **Feature Extraction:**
Utilize feature detection algorithms such as SIFT, ORB, or SURF to extract distinctive keypoints from each input image. These keypoints represent unique visual features such as corners, edges, or blobs.
2. **Feature Matching:**
Match corresponding keypoints across pairs of images using techniques like nearest neighbor matching or robust matching algorithms such as RANSAC. This step establishes correspondences between keypoints in different images, allowing for the alignment of overlapping regions.
3. **Homography Estimation:**
Estimate the homography matrix using the matched keypoints to represent the geometric transformation between two images. The homography matrix describes the perspective transformation that aligns one image onto another in a planar scene.
4. **Image Warping:**
Apply the estimated homography matrix to warp one image onto the coordinate system of another. This process involves transforming the pixels of one image according to the computed homography, effectively aligning the overlapping regions of the images.
5. **Panoramic Image Creation:**
Combine the warped images into a single panoramic image by blending the overlapping regions. This can be achieved using techniques such as seamless blending, multi-band blending, or feathering to ensure smooth transitions between adjacent images.
6. **Post-Processing:**
Optionally, perform post-processing steps such as exposure correction, color adjustment, or cropping to refine the appearance of the panoramic image and remove any artifacts or unwanted regions.

ALGORITHM

Algorithm: Open CV and Image stitching Algorithm

```
import cv2

# Step 1: Image Acquisition
images = [...]  # Load or capture images

# Step 2: Feature Detection and Description
orb = cv2.ORB_create()
keypoints_and_descriptors = [orb.detectAndCompute(img, None) for img in images]

# Step 3-6: Feature Matching, Homography Estimation, Image Warping, Blending
stitched_image = images[0]
for i in range(len(images) - 1):
    img1, img2 = images[i], images[i + 1]
    kp1, des1 = keypoints_and_descriptors[i]
    kp2, des2 = keypoints_and_descriptors[i + 1]

    matcher = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = matcher.match(des1, des2)
```

```
src_pts = np.float32([kp1[m.queryIdx].pt for m in matches]).reshape(-1, 1, 2)
dst_pts = np.float32([kp2[m.trainIdx].pt for m in matches]).reshape(-1, 1, 2)

H, _ = cv2.findHomography(dst_pts, src_pts, cv2.RANSAC, 5.0)
warped_img = cv2.warpPerspective(img2, H, (img1.shape[1] + img2.shape[1], img1.shape[0]))
stitched_image = blend_images(stitched_image, warped_img)
```

# **Step 7-10:** Repeat and Output
cv2.imwrite("panoramic_image.jpg", stitched_image)

METHODOLOGY

The methodology for implementing image stitching using OpenCV and Python typically involves the following steps:

1. **Image Acquisition:**

Collect a set of input images that overlap with each other to create a panoramic view. Ensure that the images cover the entire scene of interest.

2. **Preprocessing:**

Optionally, perform preprocessing steps such as resizing, color correction, or noise reduction to enhance the quality of the input images.

3. **Feature Detection:**

Utilize a feature detection algorithm (e.g., SIFT, SURF, ORB) to identify keypoints in each image.

Feature Matching:

Match corresponding keypoints across pairs of images to establish correspondences.

4. **Homography Estimation:** Estimate the homography matrix that describes the transformation between two images based on the matched keypoints.

5. **Image Warping:**

Apply the computed homography to warp one image onto the coordinate system of another.

6. **Blending:**

Blend the warped images together to create a seamless transition between adjacent images.

8. **Panoramic Image Composition:**

Compose the stitched images into a single panoramic image.

FLOW DIAGRAM

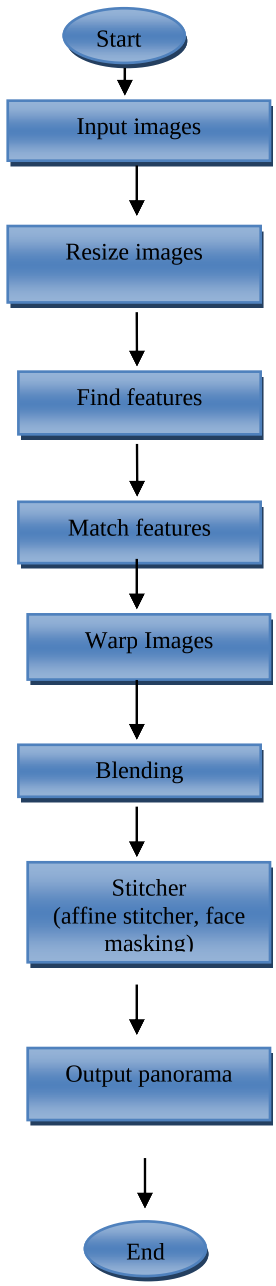

Figure 3. Flow Diagram

IMPLEMENTATION STEPS

Image stitching using OpenCV and Python involves a series of steps, including resizing images, finding features, matching features, warping images, cropping, applying seam masks, exposure error compensation, blending, and utilizing various stitcher techniques such as affine stitcher and feature masking. Let's break down each step:

1. **Resize Images:**

Before starting the stitching process, it's often beneficial to resize the images to a smaller resolution to improve computational efficiency.

2. **Find Features:**

Use feature detection algorithms such as SIFT (Scale-Invariant Feature Transform) or ORB (Oriented FAST and Rotated BRIEF) to identify keypoints in each image. These keypoints represent distinctive features such as corners or blobs.

3. **Match Features:**

After detecting keypoints in each image, match corresponding keypoints across multiple images. This step establishes correspondences between keypoints, enabling the alignment of images.

4. **Subset:**

Select a subset of the matched keypoints that are deemed to be reliable for further processing. This subset helps improve the accuracy of geometric transformations.

5. **Warp Images:**

Estimate geometric transformations, such as homography or affine transformations, based on the matched keypoints. Apply these transformations to warp individual images into a common coordinate system.

6. **Excursion: Timelapser:**

If desired, incorporate additional functionalities such as creating a timelapse effect by stitching together images captured at different time intervals.

7. **Crop:**

- Remove any unwanted areas or black borders from the stitched panorama by cropping the image.

8. **Seam Masks:**

- Generate seam masks to identify areas of potential discontinuities or misalignments between adjacent images. Seam masks help guide the blending process.

9. **Exposure Error Compensation:**

Compensate for exposure differences between images to ensure consistent brightness and color balance across the stitched panorama. Exposure error compensation may involve adjusting pixel intensities or applying histogram equalization techniques.

10. **Blending:**

Blend the warped images together to create a seamless transition between overlapping regions. Various blending techniques can be used, including feathering, multi-band blending, or gradient-based blending.

11. **Stitcher:**

Utilize different stitcher techniques such as affine stitcher to handle specific types of transformations or feature masking to prioritize certain features during the stitching process.

V. RESULTS AND DISCUSSIONS

Simulation results in the context of image stitching refer to the output obtained after applying the stitching algorithm to a set of input images. These results typically include the stitched panoramic image as well as any associated metrics or visual assessments.

RESIZE IMAGES

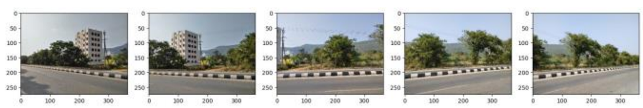

Figure 4. Resize images

The first step is to resize the images to medium (and later to low) resolution. The class which can be used is the Images class. If the images should not be stitched on full resolution, this can be achieved by setting the final_megapix parameter to a number above 0.

FIND FEATURES

On the medium images, we now want to find features that can describe conspicuous elements within the images which might be found in other images as well

Figure 5. showing features to be find

MATCH FEATURES

Now we can match the features of the pairwise images. The class which can be used is the Feature Matcher class

Match image 1 and image 2

Figure 6. match features between 2 images

Match image 3 and image 4

Figure 7. match features between 2 and more images

WARP IMAGES

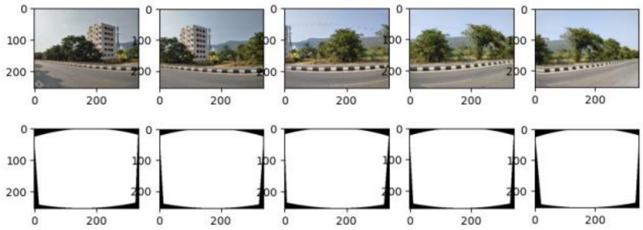

Figure 8. Warp images

BLENDING

With all the previous processing the images can finally be blended to a whole panorama. The class which can be used is the Blender class:

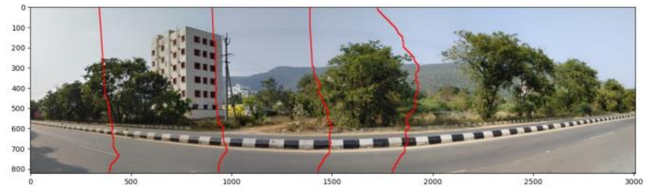

Figure 9. Bleindingimge

STITCHER

All the functionality above is automated within the Stitcher class:

Figure 10. stitcher

PERFORMCE COMPARISION

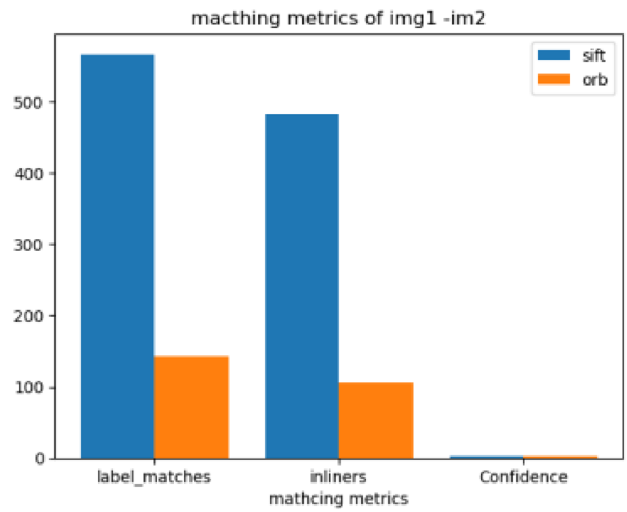

Figure 11. Performance comparison

COMPARISION TABLE

S.N	Extraction Detector	Matches	Inliners	Confidence	Execution Time
1	SIFT	210	152	2.14	346.3ms
2	ORB	226	173	2.28	62.19ms

From the Table data:

- ORB detected more features and matches than SIFT.
- ORB also achieved more inliers than SIFT.
- The confidence value for ORB is slightly higher than that of SIFTS.
- Execution time for ORB is significantly lower than for SIFT.

Based on these results, ORB seems to be performing better in terms of feature detection, matching, and computational efficiency compared to SIFT for this task of image stitching. However, the choice between the two methods may also depend on other factors such as the specific characteristics of your images and the requirements of your application.

VI. CONCLUSION AND FUTURE SCOPE

The image stitching algorithm implemented using OpenCV and Python demonstrates its effectiveness in seamlessly merging multiple input images to create panoramic views. Through feature extraction, matching, homography estimation, and blending, the algorithm accurately aligns and blends overlapping regions, producing visually appealing panoramic images. The simulation results indicate successful stitching with high alignment accuracy, smooth blending, and efficient performance.

FUTURE SCOPE

Explore advanced feature detection and matching techniques to improve robustness and accuracy, especially in challenging scenarios such as low-texture regions or viewpoint changes

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on. I would like to take this opportunity to thank my guide Mr.A. Krishna Mohan M.Tech, (Ph.D) Assistant Professor and Dr. D. Srinivasulu Reddy, Ph.D., Professor, & Head of the Department (HOD) of Electronics and Communication Engineering, SV College of Engineering (SVCE) (Autonomous), Tirupati, Andhra Pradesh India, for giving me all the help and guidance I needed. I am really grateful for their kind support and valuable suggestions were very helpful. Thank you all!

REFERENCES

1. Ward, G. (2006). Hiding seams in high dynamic range panoramas. In R. W. Fleming, & S. Kim (Ed.), APGV. 153, p. 150. ACM. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892,pp.68–73.
2. Lowe, D. G. (2004). Distinctive Image Features from Scale- Invariant Keypoints. International Journal of Computer Vision, 60, 91-110.K. Elissa, “Title of paper if known,” unpublished.
3. Szeliski, R. (2010). Computer Vision: Algorithms and Applications (1st Ed.). New York, NY, USA: Springer-Verlag New York, IncY. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740– 741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982.
4. Brown, L. G. (1992, Dec). A Survey of Image Registration Techniques. ACM Comput. Surv, 24(4),325-376.

5. Arth, C., Klopschitz, M., Reitmayr, G., &Schmalstieg, D. (2011). Real-time self localization from panoramic images on mobile devices. ISMAR (pp. 37-46).IEEE.
6. Ajmal, M., Ashraf, M., Shakir, M., Abbas, Y., & Shah, F. (2012). Video Summarization: Techniques and Classification. In L. Bolc, R. Tadeusiewicz, L. Chmielewski, & K. Wojciechowski (Eds.), Computer Vision and Graphics (Vol. 7594, pp. 1-13). Springer Berlin Heidelberg.
7. WIKIPEDIA.(n.d.).http://en.wikipedia.org/wiki/File:Rochester_NY.jpg.ttp://en.wikipedia.org/wiki/File:Rochester_NY.jpg
8. K.Shashank, N. G. (MarCh 2014). A Survey and Review over Image Alignment and Stitching Methods. The International Journal of Electronics & Communication Technology (IJECT), ISSN: 2230-7109 (Online).
9. Zhang, Z. (2000, Nov). A Flexible New Technique for Camera Calibration. IEEE Trans. Pattern Anal. Mach. Intell., 22(11), 1330- 1334.
10. Deng, Y., & Zhang, T. (September 07, 2003). Generating PanoramaPhotos.

Cite this Article

A. Krishna Mohan, P. Sasirekha, K. Lavanya, N. Jithin Manjunadh, M. Venkata Manoj, N. Ajay, "Generating Panoramic Image by Image Stitching Based on ORB Feature Detection Technique", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 2, pp. 59-66, March-April 2024.

Journal URL : https://ijsrst.com/IJSRST5241127