IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 14, Issue 5, Ser. I (Sep.-Oct. 2019), PP 56-67
www.iosrjournals.org

Panoramic Image Creation

Sruthi P. [1], Sruthi Dinesh[2]

[1] *P. G. Student, Electronics and Communication, GCEK, Kerala, India*

[2] *Asst. Professor, Electronics and Communication, GCEK, Kerala, India*

Abstract: *This paper presents an effective method in panoramic image creation. Here an automatic method of feature detection and feature mating using SIFT (Scale Invariant feature transform) algorithm has been used. SIFT algorithm is very robust method that can detect and describe local features in the image. Then find the overlapping area of these two images. A minimal cost path in the overlapping area of two images is found by dynamic programming method. This minimal-cost path (optimal seam) is used to stitch images. Dynamic programming is faster than other seam finding methods and uses little memory. The overlapping images are cut along the seam and merge together. Here there are seven steps used in image stitching which include: input image, feature detection, feature matching, image registration, computing homography using RANSAC, image warping, and finally image labeling using optimal seam.*

Key Words: *SIFT, RANSAC, Optimal seam, Dynamic programming, Image stitching*

Date of Submission: 13-11-2019 Date of Acceptance: 27-11-2019

I. Introduction

Panoramic image is created by stitching a series of overlapping images. This paper aim is to create a Matlab script that will stitch two or more images together to create one larger image. Field of view (FOV) of our normal consumer camera is 50^0 x 35^0 . FOV of human vision is 176^0 x135^0. FOV of panoramic image is up to 360^0 x180^0 (ie.360^0 horizontal and 180^0 vertical field of view). Here panoramic image is created by following six steps. They are image accusition, feature detection, feature extraction2, homography calculation, image warping and image labeling.

Panoramic Image Stitching Techniques

There are two main techniques considered for image stitching techniques in which first one is regional and second one is feature based techniques. Approaches that use pixel-to-pixel matching are often called direct methods as opposed to the feature-based methods described in this section.

Direct Methods

The direct technique depends on comparing all the pixel intensities of the images with each other. Direct techniques minimize the sum of absolute differences between overlapping pixels or use any other available cost functions. These methods are computationally complex as they compare each pixel window to others. They are not invariant to image scale and rotation. The main advantage of direct methods is that they make optimal use of the information available in image alignment. They measure the contribution of every pixel in the image. However, the biggest disadvantage of direct techniques is that they have a limited range of convergence.

Feature Based Methods

The simplest way to find all corresponding feature points in an image pair is to compare all features in one image against all features in the other using one of the local descriptors [3]. For image stitching based on feature-based techniques, feature extraction, registration, and blending are the different steps. Feature-based methods begin by establishing correspondences between points, lines, edges, corners or other geometric entities. Characteristics of robust detectors include invariance to image noise, scale invariance, translation invariance, and rotation transformations. There are many feature detector techniques, such as Harris, Scale- Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Features from Accelerated Segment Test (FAST), PCA-SIFT and ORB techniques.

II. Proposed System

The proposed method uses a combined feature based and optimal seam finding method for panoramic image creation. Steps to create panorama using this method is described below.
Step 1: SIFT algorithm is used for feature extraction and feature matching.
Step 2: RANSAC algorithm is used to find homography matrix and eliminating outliers (wrong matching points). Geometric distortion due to change of camera orientation can be compensated using this homography matrix.
Step 3: Next stage is image warping, which includes correcting the geometric distortion of an image that is warp the image pixels from both the image on 2D plane using homography matrix.
Step 4: After image warping extract the overlapping area. Step 5: Next stage is image labeling. Here optimal seam finding method is used for image labeling. Dynamic programming method used for finding optimal seam in the overlapping area. Then cut the image along the seam and stitch together.
Block diagram of this method is given in fig -1. So this panoramic image stitching model consists of six stages: input images (image acquisition), features detection, matching feature, computing homography, image warping, image labeling. In the following subsections, these stages of feature based panorama image stitching will be described in detail.

Input Image (Image Acquisition)

The first stage of image stitching is the image acquisition stage. Image acquisition can be broadly defined as the action of retrieving an image from some sources. Typically, images can be acquired for panoramic imaging by a hand-held camera. Input image is a sequence of overlapping images.

SIFT (Scale Invariant Feature Transform) is used here, so the y-axis shift in source images is not a matter. Also by using SIFT the stitching order and any rotation change would not be considered.

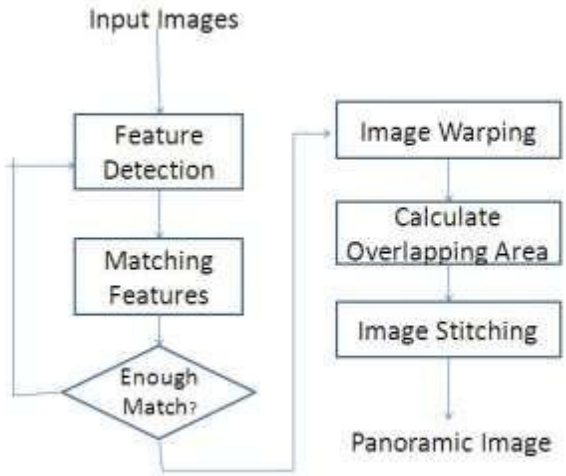


Fig -1: Block Diagram of Image Stitching

Feature Detection

The second step in image stitching process is the feature detection which is considered as the main image stitching stage. Features can be defined as the elements in two or more input images. It relies on the idea that instead of looking at the image as a whole, it could be advantageous to select some special points in the image and perform a local analysis on these ones.

SIFT Algorithm

In this paper, technique used for the feature extraction is SIFT which is used to extract invariant features which are stable in nature. Invariant features are those features of an image which do not change even after the scaling, rotation, or zooming and change in illumination of the image is done. Multiple level filtering and downsampling are the key factors of the SIFT. So the steps involved are feature detection, matching of stable features and wrapping up of features around those feature locations. All the basic steps involved in SIFT algorithm are described below.

Scale-Space Extrema Detection

This is the most time consuming stage. scale space extrema detection means finding the feature points (key points). Steps to detect key points are given below.

1. In this stage the input image will apply many Gaussian filters with different variance. In fig -2 five differentially blurred images are shown.
2. Take Difference of Gaussian (DoG). That is find the difference of these differentially smoothed image (pixel by pixel).
3. Find extrema. For this compare a pixel (x) with 26 pixels in current and adjacent scales(circle)(fig -2(b)). Select this pixel (x) as the keypoint if this pixel value is larger or smaller than all 26 pixels.

Discarding Low-contrast Key Points and Eliminating Edge Responses
After finding the feature location we eliminate key point which will produce unstable feature.

3.2.1.3 Orientation Assignment

In this step, our task is computing these feature points (keypoints) orientation. Orientation of feature point helps us to achieve rotation invariant. For each pixel compute the central derivative and find the gradient magnitude and direction of L (smooth image) at the scale of keypoint *(x, y).*

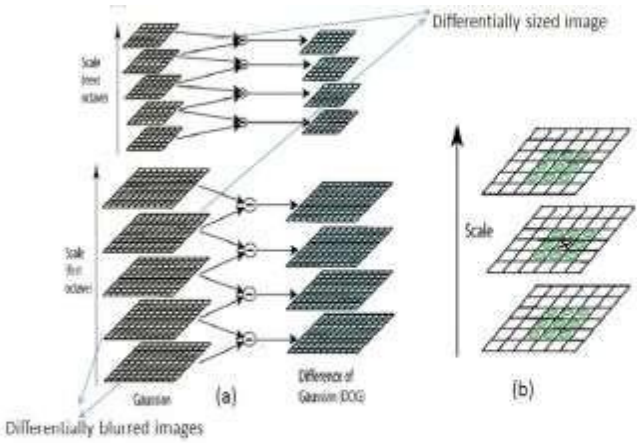


Fig -2: (a) Gaussian and DoG. (b) Checking of each keypoint with its 26 neighbors

$$(\ ) \quad \sqrt{(\ (\quad) \ (\quad))^{} \ (\ (\quad) \ (\quad))} \qquad (1)$$

$$(\quad) \qquad \frac{(\ (\quad) \ (\quad))}{(\ (\quad) \ (\quad))} \qquad (2)$$

Equation (1) gives the magnitude and (2) gives the direction of the keypoint(x,y). Around the keypoint look at the orientation (direction)of each of the pixel. Then create a weighted direction histogram in a neighbourhood of the keypoint. Fig -3 shows an example of direction histogram. Select the peak value in the histogram as direction of the keypoint

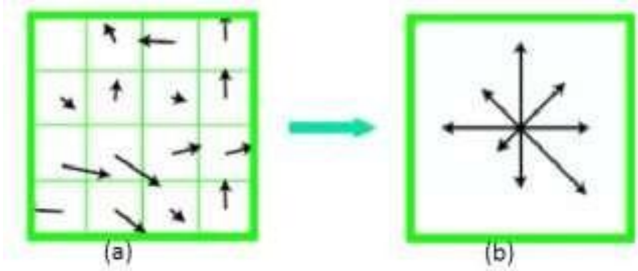


Fig -3: (a)Direction of neighborhood pixels of a keypoint .
(b) Direction histogram..

2.2.1.4 Key Point Descriptor

For each key point and orientation pair we compute their SIFT descriptor using local image gradient magnitude and angle. Extraction of descriptors at keypoint, compute relative orientation and magnitude in 16 16 neighborhood at key point. Divide the 16 16 into 4 4 blocks. So we have 16 blocks. Form direction histogram for each of the 4 4 block. each histogram is 8 dimensional. then we concatenate 16 histogram of 8 dimension into one long vector of 128 dimension(16 8). Therefore SIFT descriptor is a 128 dimensional vector. Each key point will collect data from a 128 128 pixel square and store the result ina 128-dims vector. The image is stored in global memory so is the vector. All the operations require a lot of memory bandwidth, resulting in low instruction throughput. Our solution is to break the computation of the 128-dims vector into many different threads. Each thread will compute 8 entries of the 128-dims vector. So we can store the vector data in 8 registers instead of global memory. Although we need to compute the same information in different threads, the overhead is small compared to the gain of reduced memory access.

Matching Features

Key points between two images are matched by identifying their nearest neighbors. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminates around 90% of false matches while discards only 50% correct matches.

Image Registration

Image registration is the task of matching two or more images. It has been a central issue for a variety of problems in image processing such as object recognition, monitoring satellite images, matching stereo images for reconstructing depth, matching biomedical images for diagnosis, etc. Registration is also the central task of image stitching procedures. Carefully calibrated and rerecorded camera parameters may be used to eliminate the need for an automatic registration. User interaction also is a reliable source for manually registering images (e.g. by choosing corresponding points and employing necessary transformations on screen with visual feedback). Automated methods for image registration used in image stitching literature can be categorized as follows: Feature based [7] methods rely on accurate detection of image features. Correspondences between features lead to computation of the camera motion which can be tested for alignment. In the absence of distinctive features, this kind of approach is likely to fail. Exhaustively searching for a best match for all possible motion parameters can be computationally extremely expensive. Using hierarchical processing (i.e. coarse-to-fine [5]) results significantly in speed-ups. We also use this approach taking advantage of parallel processing for additional performance improvement. Frequency domain approaches for finding displacement and rotation/scale are computationally efficient but can be sensitive to noise. These methods also require the overlap extent to occupy a significant portion of the images (e.g. at least 50%). Iteratively adjusting camera-motion parameters leads to local minimums unless a reliable initial estimate is provided. Initial estimates can be obtained using a coarse global search or an efficiently implemented frequency domain approach.

Computing Homography

A 2D point (x, y) in an image can be represented as a 3D require a set of correspondences as input. So far these algorithms are only robust with respect to noise if the source of this noise is in the measurement of the correspondence feature positions. There will be other situations where the input will be corrupted with completely false correspondences, meaning that the two features in the images don't correspond to the same real world feature at all. There is a need to discuss ways to distinguish inlier and outlier correspondences so that the homography can be estimated robustly using only inlier matches.

RANSAC Algorithm

RANSAC (Random Sample Consensus) is the most commonly used robust estimation method for homographies. The idea of the algorithm is pretty simple; for a number of iterations, a random sample of 4 correspondences is selected and a homography H is computed from those four correspondences. Each other correspondence is then classified as an inlier or outlier depending on its concurrence with H. After all of the iterations are done, the iteration that contained the largest number of inliers is selected. H can then be recomputed from all of the correspondences that were considered as inliers in that iteration. One important issue when applying the RANSAC algorithm described above is to decide how to classify correspondences as inliers or outliers. Statistically speaking, the goal is to assign a distance threshold, t, (between x0 and HX for example), such that a probability point is an inlier. Hartley and Zisserman [12] provide a derivation of how to calculatet. Another issue is to decide how much iteration to run the algorithm for. It will likely be infeasible to try every combination of 4 correspondences, and thus the goal is to determine the number of iterations, N, that ensures with a probability p that at least one of the random samples will be free from outliers. Then, the algorithm:

1. Selects N data items at random.
2. Estimates parameter x.
3. Finds how many data items (of M) fit the model with parameter vector x within a user given tolerance.

Call this K.

4. If K is big enough, accept fit and exit with success.
5. Repeats 1.4 L times.
6. Fails if you get here.

How big K has to be depends on what percentage of the data we think belongs to the structure being fit and how m*a*ny —

vector x=(x1, x2, x3) where and . This is called the homogeneous representation of a point and it lies on the projective plane P. Homography is an invertible mapping of points and lines on the projective plane P. Other terms for this transformation include collineation, projectivity, and planar projective transform. First for each picture, we consider 6 images that have the greatest number of feature matches to it, and then we use RANSAC to select a set of inliers that are compatible with a homography between the images. Hartley and Zisserman [12] provide the specific definition that a homography is an invertible mapping from P2 to itself such that three points lie on the same line if and only if their mapped points are also collinear. They also give an algebraic definition by proving the following theorem: A mapping from P P is a projectivity if and only if there exists a non-singular 3 3- matrix H such that for any point in P represented by vector x it is true that its mapped point equals Hx. This tells us that in order to calculate the homography that maps each xi to its corresponding xi. It is sufficient to calculate the 3 3 homography matrix, H. All of the homography estimation algorithms that are discussed

redo RANSAC.

Image Warping

Image Warping is the process of digitally manipulating an image such that any shapes portrayed in the image have been significantly distorted. Warping may be used for correcting image distortion as well as for creative purposes (e.g., morphing). While an image can be transformed in various ways, pure warping means that points are mapped to points without changing the colors. This can be based mathematically on any function from part of the plane to the plane. If the function is injective the original can be reconstructed. If the function is a bijection any image can be inversely transformed. The last step is to warp and blend all the input images to an output composite mosaic. First we need to make out the output mosaic size by computing the range of warped image coordinates for each input image. As described earlier we can easily do this by mapping four corners of each source image forward and computing the minimum x, minimum y, maximum x and maximum y coordinates to determine the size of the output image.

Finally x-offset and y-offset values specifying the offset of the reference image origin relative to the output panorama needs to be calculated. The next step is to use the inverse warping as described above for mapping the pixels from each input image to the plane defined by the reference image, to perform the forward and inverse warping of points, respectively.

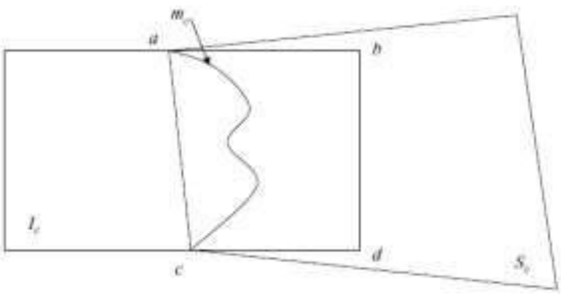


Fig -4: Finding the optimal seam between the current composite image and the current source image.

Image Labeling with dynamic programming method

Dynamic Programming method is to find optimal seams to merge the source images together quickly and using little memory, so that it can be applied for producing high- resolution panoramic images on mobile devices. We want to merge the images on places where they differ the least. As shown in fig-4, suppose that abcd is the overlapping area between the current composite image and the current source image . and are the overlapping images in the area abcd of and respectively. We compute squared differences *d* between and as an error surface,

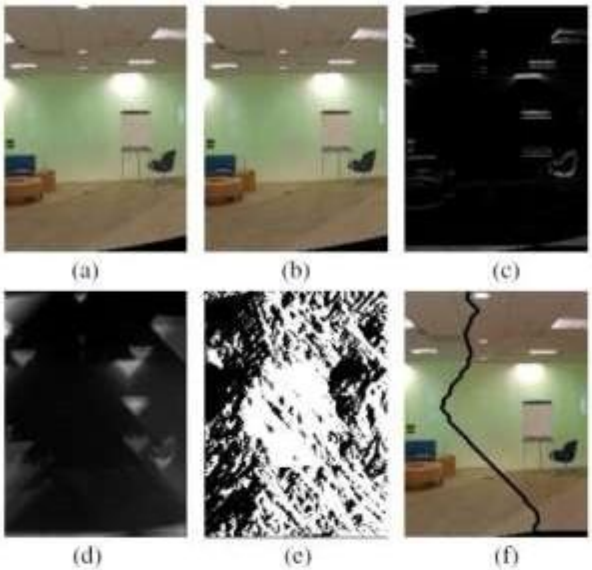


Fig -5: Process of finding the optimal seam with dynamic programming optimization.

We update the current composite image by merging the current image with the labeling information and continue the labeling process with the next source image. After all source images are processed, we obtain the final composite image.

III. Result And Discussions

The proposed system for digital image enhancement is simulated using MATLAB R2012b on a PC with a Windows 7 professional operating system.

3.1 Results of Combined Direct and Dynamic Programming Method for Image Stitching

( ) (3)

We apply dynamic programming to find a minimal cost path through this surface. We scan the error surface row by row and compute the cumulative minimum squared difference D for all paths,

( )

( ) (( ) ( ) ( ))

(4)

where h = 2,….., and w = 2,….., are the indices of the row and column of the error surface respectively. The optimal path mc can be obtained by tracing back the paths with a minimal cost from bottom to top.

For the last row, the minimum value can be used to determine the end ( ) of the optimal path. For the next upper row, if ( ) = ( ) ( )

, , then the position of the optimal path in this row is ( ). We repeat the process until all rows have been traced.

Fig -5 shows the process of optimal seam finding with dynamic programming optimization. Fig -5 (a) and (b) are the overlapping images in the overlapping area of abcd of and respectively. The error surface shown in fig 5 (c) is computed through the squared differences between the two images shown in fig-5 (a) and (b). After that, the cumulative squared difference D is computed with the error surface d and is shown in fig- 5 (d). Meanwhile, we also obtain all possible paths shown in fig-5 (e). After tracing back with dynamic programming, we obtain an optimal path shown in fig -5 (f) along which the two images in (a) and (b) match best. We use the optimal path as an optimal seam to create labeling and cut the overlapped images.

This method uses direct method (pixel to pixel matching) for finding overlapping area and dynamic programming method for image labeling. Here the Y axis shift in source images are not allowed that is images are taken using a tripod.

But it is faster ad use little memory and can create upto panoramic image. Fig -6 are source images that are overlapping and fig -7 is the overlapping area of these images extracted by direct method. Fig -8 shows the error surface and (b) shows the accumulated energy surface. Optimal path using dynamic programming shown in fig- 9.

The final panoramic image (stitched image) is shown in fig -10

Fig -6 : Source Images

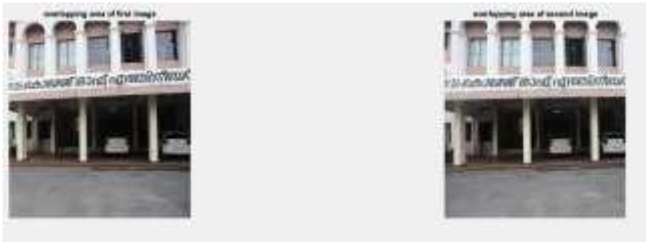


Fig -7: Overlapping Area of Source Images

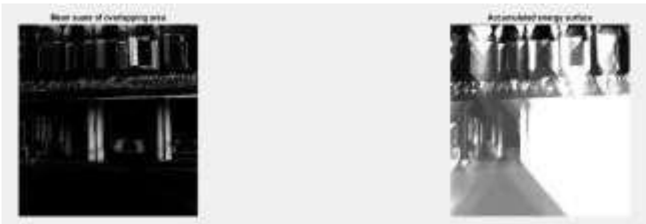


Fig -8: Error Surface and Accumulated Energy Surface

Fig -9: Blue Line Showing the Seam in the Overlapping Area

Fig -10: Stitched Image

3.2 Results of Combined Feature Based and Dynamic Programming Method for Image Stitching

This method uses feature based method for finding overlapping area and dynamic programming method for image stitching. Here Y axis shift in source images are allowed, which is more time consuming. Seam carving using dynamic method reduce stitching artifacts. This algorithm is suited for both horizontal and vertical stitching. Fig - 19 shows the horizontal stitching using this algorithm (source images and stitched image). Fig -17 shows the vertical stitching (source images and stitched image). Fig- 21 shows the stitching of 2 dimensional sweep images. Fig- 11 shows the feature extracted using scale invariant features and blue line shows matching of these points. The current composite image and transformed image using homographic matrix is shown in fig- 12(a) and (b).

Figure shows the overlapping area of the two images. White area in fig-13 is the overlapping area of the images. Then we extract the overlapping area from this image fig-13 shows this overlapping region of these images. Error surface (square difference overlapping area of these images) cumulative error surface (cumulative square difference) are shown in fig-14. Then calculate the minimal cost path ie optimal path in the overlapping area. Red line showed in the fig-15(a) optimal seam. Fig-15(b) shows the merged overlapping area. Figure 4.11 shows the final panoramic image using our algorithm. Fig-17, 18, 19, 20, 21 shows results of this algorithm for different images. Fig-17 shows vertical stitching. Fig-18, 19, 20 are shows output of different horizontal stitching. Fig-21 shows both horizontal and vertical stitching.

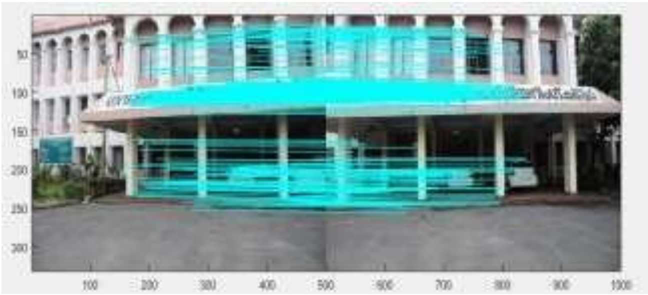


Fig -11: Feature Extracted Using SIFT

Fig -12: (a)Current Composite Image (b)Transformed Image

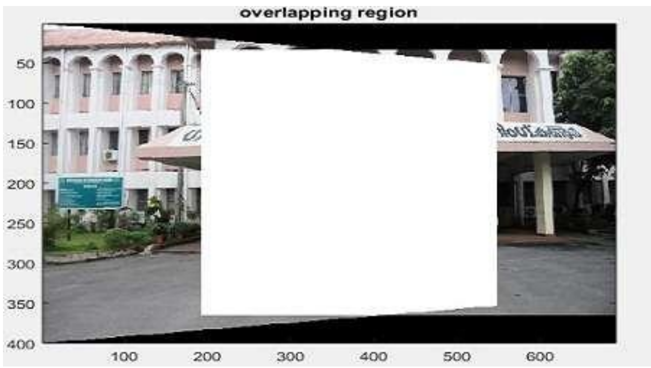
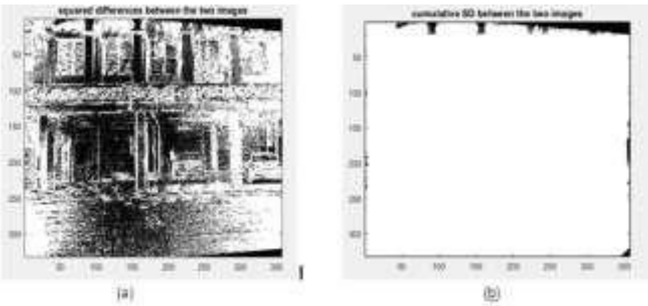


Fig -13: Finding Overlapping Area(white color)



ig -14: (a)Square Difference Between the Two Images (b)Cumulative Square Difference Between the Two Images

Fig -15: (a)Optimal Seam in the Overlapping Area (b)Merged Overlapping Region

Fig -16: Stitched Image

Fig-17: (a)(b)(c)Source Images (d)Stitched Image

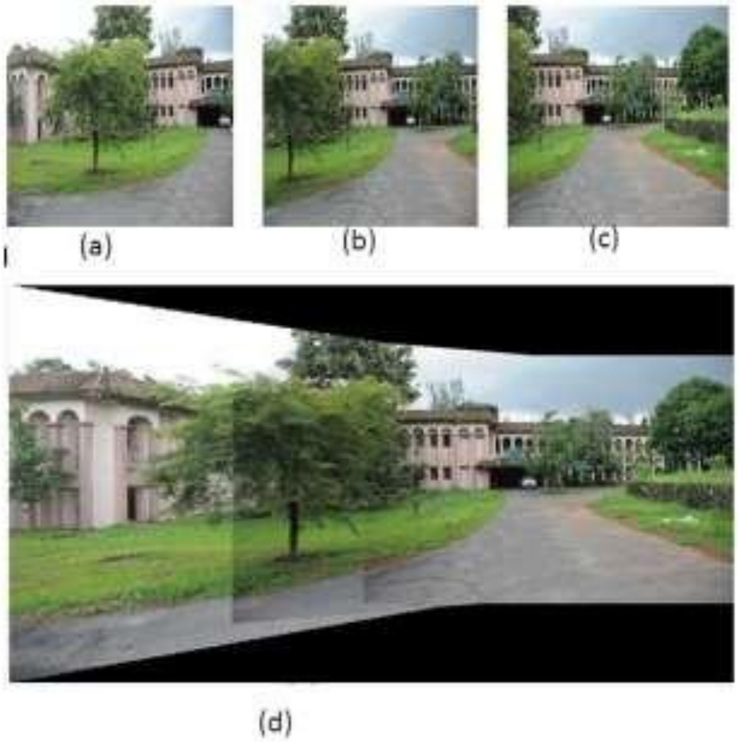


Fig -18. (a)(b)(c)Source Images (d) Stitched Image

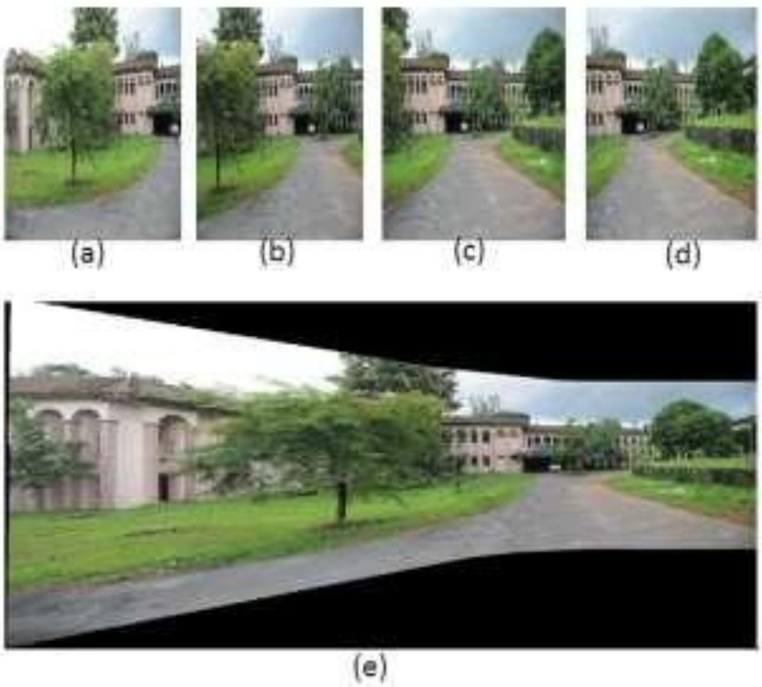


Fig -19. (a)(b)(c)(d)Source Images (e) Stitched Image

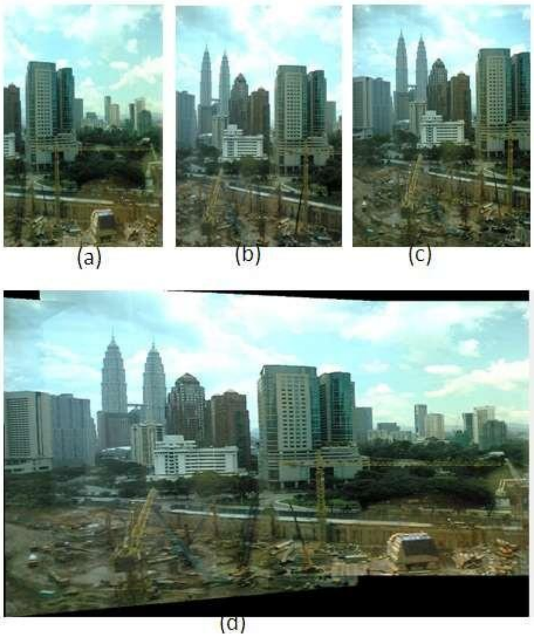
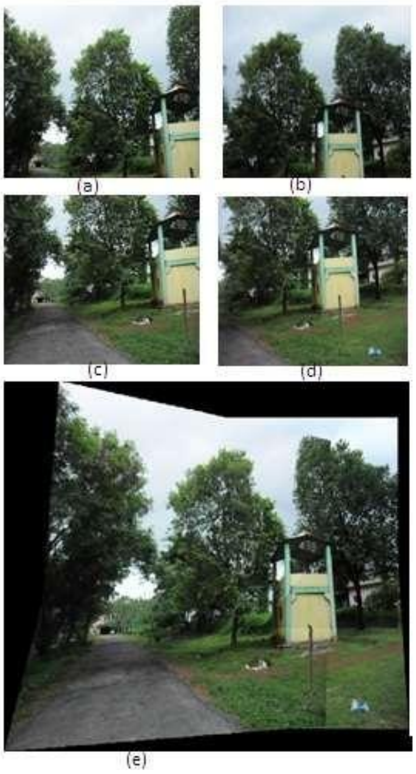


Fig -20. (a)(b)(c)Source Images (d) Stitched Image

IV. Conclusions

Image stitching is useful for a variety of tasks in vision and computer graphics. Due to the wide range of applications, image stitching is one of the important research areas in the field of image processing. Here we have presented some of the very fundamental and basic techniques and improved algorithms used in image stitching. This paper presents a complete process for image stitching. Combined dynamic programming and SIFT algorithm give better result than other methods and give less stitching artifacts. Using this proposed method stitch images that are taken randomly. Images with Y-axis shift can stitch using this method. Horizontal, vertical, and both (horizontal and vertical) stitching are possible using this algorithm and gives better results than existing methods.



. **Fig -21**. (a)(b)(c)(d)Source Images (e) Stitched Image

References

[1]. Cheng-Ming Huang,Shu-Wei Lin,Jyun-Hong, “Efficient Image Stitching of Continuous Image Sequence With Image and Seam Selections,” IEEE Sensor Journal, vol.15,2015.
[2]. Wang M, “Panorama Painting: With a Bare Digital Camera,” ICIG’09. Fifth International Conference, 2009.
[3]. Brown M, Lowe DG, “Automatic panoramic image stitching using invariant features,” Int J Comput Vis, Vol.1,59-73,2007.
[4]. Kekre HB, Thepade SD, “Rotation invariant fusion of partial image parts in vista creation using missing view regeneration,” WASET Int J Electr Comput Eng Syst, 47:660,2008.
[5]. Helmut D, “Panorama Tools. Open source software for immersive imaging international VR photography conference,” http://webuser.fhfurtwangen.de/ dersch/IVRPA.pdf, June 2007.
[6]. Wen-Yan L, Siying L, Yasuyuki M, Tian-Tsong N, Loong-Fah C , “Smoothly varying affine stitching. Computer vision and pattern recognition (CVPR),”
[7]. IEEE Conference, Providence, RI, 2011.
[8]. Song B, Yongqing F, Wang J, “Automatic panorama creation using multi-row images,” Inf Technol J, 2011.
[9]. Xiong Y, Pulli K, “Fast image stitching and editing for panorama painting on mobile phones,” In: IEEE Comput Soc Conf Comput Vis Pattern Recogn Workshops (CVPRW), 2010.
[10]. D.Lowe, “Distinctive image feature from scale invarient key points,”International Journal of computer vision, 2004.
[11]. Rachumallu Balaramakrishna, N Swetha, Pillem Ramesh, “Creating Image Panoramas Using Homography Warping,” International Journal of Engineering
[12]. Trends and Technology (IJETT), vol.4, April 2013.
[13]. Hyung Il Koo, Nam Ik Cho, “Feature-based image registration algorithm for image stitching applications on mobile devices,” IEEE Transaction on Consumer Electronics, Vol.57,2010.
[14]. Richard Hartley, Andrew Zisserman, “Multiple View Geometry in Computer Vision, Second Edition,” Cambridge University Press, 2004.

Sruthi P. "Panoramic Image Creation." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 14.5 (2019): 56-67.