©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

COMPREHENSIVE ANALYSIS AND EFFICIENCY COMPARISON OF IMAGE STITCHING TECHNIQUES

Raghav Mehrish[1], M. Jasmine Pemeena Priyadarsini[1], K. Murugesan[2], Srinivasa Rao Inbathini[1], A. Jabeena[1] and S. Revathi[1]
[1]School of Electronics Engineering, VIT University, Vellore, Chennai, India
[2]Sree Sastha Institute of Engineering and Technology, Chennai, India
E-Mail: raghav297@gmail.com

ABSTRACT

Panorama stitching or image stitching is gaining prominence in today's world due to its applications in wide context aware imaging. The project aims on three different image stitching techniques and comparing their efficiencies and computational time. After the image is acquired, if the image parameters are unknown, feature detection and extraction is carried out without undistortion and cylindrical warping. In this case, the SIFT features are calculated from the acquired images and the images are translated based on the obtained homography. Then images are blended using alpha or weighted blending. If the image parameters, such as focal length and intrinsic parameters are known after image acquisition, the image first undergoes undistortion and warping. After that, two approaches are followed to compare efficiencies between the methods. The first approach involves the direct method in which the translation between the images is calculated with the need for features to reduce the calculated time manifolds and aid in faster translation approximation. And also feathering is performed to blend the images seamlessly and obtain the stitched image. In the second approach, the feature based method is used wherein the SIFT features are calculated for the image sets. Following this, the features are used to calculate the homography between the two images and determine the translation. Then, the images are stitched using a different blending technique called pyramid blending. This technique proves slightly more efficient than the previous technique due to computation of Laplacian and Gaussian pyramids of the images.

Keywords: image undistortion, warping, feature detectors, homograpyh, aforementioned stitching Laplacian and Gaussian, RANSA, key point localization, original Laplacian image, compensation, fourier transform.

1. INTRODUCTION

Signal processing is a specialization of electrical engineering and mathematics that analyzes signals and operates on them, either in discrete or continuous time, performing useful analysis on those signals. Some of the most common signals can include sound, images, sensor data, time-varying measurement values, telecommunication transmission signals, control system signals and many others. These signals are analog or digital electrical representations of time-varying or spatial-varying physical magnitude.

Depending on the kind of signal, there are three types of signal processing analog signal processing for not digitized signals, as radio, telephone, radar, and television systems; discrete time signal processing for sampled signals that are defined only at discrete points in time; and digital signal processing as the processing of digitized discrete time sampled signals, done by computers or specialized digital signal processors.

Digital signal processing is aimed at measuring, filtering and compressing continuous real analog signals. Its first step is to convert the signal from an analog to a digital form, sampling it using an analog-to-digital converter, which converts the analog signal into a stream of numbers. Digital signal processing has many advantages over analog processing in many areas of applications, such as correction in transmission and error detection as well as data compression. It also includes subfields such as audio and speech signal processing, sonar and radar signal processing, spectral estimation, statistical signal processing, digital image processing, signal processing for communications, and many others.

The subcategory of signal processing analyzed is image processing, also usually refers to digital image processing. Image processing is a type of signal processing where the input is an image, such as a photograph or video frame and the output of image processing can be an image or a set of parameters related to the image. Most image processing techniques involve treating images as a two dimensional signals and applying standard signal processing techniques. Usually, digital image processing uses a wide range of computer algorithms to perform image processing on digital images, avoiding problems such as the increase of noise and signal distortion during the process. Medical and microscope image processing face and feature detection, computer vision and image stitching are some of the different applications in the field of image processing.

1.1. Image

It is necessary to understand the basic processes that work with images before elaborating on the image stitching process. There can be differences between still image (digital image) and moving image (digital video). In this section, the characteristics and different formats of digital image are explained. Following which a brief overview considering video stitching as the next step of image stitching, a brief introduction to the digital video is presented.

1.1.1. Still image

Still images can be manipulated by computers which is why they are used a numeric representation of a two-dimensional static image, known as digital image. To obtain a digital image from an analog source, the digitalization process is performed in some devices such as scanners or digital cameras. After that, the digital image is prepared to be processed by processes such as camera calibration. There are different digital image formats to work such as bitmap or raster format and vector. Often, it can combine both formats in one image.

1.1.2. Raster image

Raster graphic image or bitmap is composed by a serial of points, called pixels, that contains colour information. Bitmap images depend on the resolution, containing a fixed number of pixels. Each pixel has a concrete location and colour value information, what convert the pixel to the basic information unit of the image. The pixels are distributed creating a grid of cells, where each cell is a pixel, and all together build the whole image. When the grid is modified, it affects the distribution, number and colour information of every pixel, and therefore, the final image. The image resolution is the number of pixels shown per longitude unit printed in an image, normally in pixel per inch. The quality of a bitmap image is determined in the moment of its creation, so it cannot amplify its resolution without modifying the image, normally deforming it and losing definition. This resolution is proportionally related to the file size more pixels imply more bits.

There are two types of compression namely lossy and lossless techniques. The first one compresses keeping image details and colour information, while the second deletes both.

Some common bitmaps formats are:

- GIF - Graphical Interchange Format: It works with two methods to compress, CLUT (Colour Look-Up Table) and LZW (explained below). It is one of the most used, especially in web images, because it offers more possibilities and higher level compression than others.
  Suitable for 256 colours compression.
- RLE - Run Length Encoding: Lossless compression technique that registers a single colour value for a group of pixels with the same colour. This technique is exploited in bitmaps images with a large amount of equal colours, saving a lot of weight and keeping the quality.
- LZW - Lempel-Ziv-Welch: Similar operation to RLE.
- JPEG - Join Photographic Expert Group: It is one of the most adequate formats for images with more than 256 colours, appropriate for colour photos and web images compression.
- PNG - Portable Network Graphics: It has become important over the last times. It allows lossless compression, merging perfectly any image edge with the background.
- BMP - BitMaP: It is the Windows format, very popular but its compression is poor compared with other formats such as JPEG.
- PSD - Photo Shop Document: It is the format for the Adobe program, widely used because it is one of the most powerful photography programs graphically.
- TIFF - Tag Image File Format: It is admitted in almost all the edition and image applications. It allows many possibilities for both Mac and PC.

1.1.3. Vector image

Vector images or vector graphics oriented to objects are made by vectors, objects mathematically created. Each vector is defined by a serial of points that have some handles to control the line shape created between them. The curve is totally defined by nodes or anchor points, and the handles.

Some of the most popular and used vector graphics formats are:

- CDR - Corel DRaw: Format generated by the program with the same name.
- AI - Adobe Illustrator: Characteristics similar to Corel DRaw.
- EPS - Encapsulated PostScript: Very adaptable format. It is one of the best formatsto be imported from most of design software.
- WMF - Windows MetaFile: Format developed by Microsoft, and especially suited towork with Microsoft programs.

1.1.4. Moving image

A moving image is typically a movie (film), or video, including digital video.

Specifically, digital video is composed for a series of orthogonal bitmap digital images displayed in rapid succession at a constant rate. In the context of video these images are called frames, and typically is measured the rate at which these frames are displayed in frames per second (FPS).

There are two different formats to get the images, interlaced and progressive scan.

The interlaced scan gets the image in groups of alternate lines, first the odd lines, and after the even lines, repeating progressively. In the other case, a progressive scan gets every image individually, with all scan lines being captured at the same moment in time. Thus, interlaced video captures samples the scene motion two times faster as often as progressive video does, for the same number of frames per second.

1.2. Image processing

Image processing is required to pre-process the image and convert it into a form suitable for further analysis. This chapter reviews standard image processing operators and transforms that map pixel values from one image to another.

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

1.2.1. Operators

The point operators or processes are the simplest kind of image processing transforms, where each output pixel's value depends on only the corresponding input pixel value. This can be denoted as a function that takes one or more input images f(x) and produces an output image g(x).

$$G (x) = h (f(x))$$

For sampled images, the domain consists of a finite number of pixels locations, replacing the value x = (i,j)in the equation.

Two commonly used point operators are multiplication and addition with a constant

$$g (x) = f (x)a(x) + b(x)$$

where a and b are said to control contrast and brightness, respectively. Multiplicative gain is a linear operation related to the superposition principle.

$$h (f0 + f1) = h (f0) + h (f1)$$

One highly used non-linear transform applied before further processing is gamma correction, which is used to remove the non-linear mapping.

$$g (x) = [f(x)]^{1/y}$$

Moreover, there are the colour transforms, that adding the same value to each colour channel not only increases the apparent intensity of each pixel, it can also affect the pixel's hue and saturation. This colour balancing can be performed either by multiplying each channel with a different scale factor or by more complex processes.

The automatic way to deter mine the best values of the brightness and gain controls described before is plotting a histogram of the individual colour channels and luminance values. From this distribution, we can compute relevant statistics such as the minimum, maximum and average intensity values. One common solution is to perform histogram equalization, to find an intensity mapping function such that the resulting histogram is flat. T.

Locally adaptive histogram equalization is an example of neighbourhood or local operator, which uses a collection of pixel values in the surrounding area of a given pixel to determine its final output value. In addition, neighbourhood operators can be used to filter images in order to add soft blur, sharpen details, accentuate edges, or remove noise. There are linear filtering operators that involve weighted combinations of pixels in small neighbourhoods, and non-linear filtering operators such as median or bilateral filters and distance transforms.

1.2.2. Linear filtering

The most commonly used type of neighbourhood operator is linear filter, in which an output pixel's value is determined as a weighted sum of input pixel values. The entries in the mask h (k, l) or kernel, are often called the filter coefficients. Another common variant and compactly notated formula is the convolution operator,

$$g = f*h$$

and h is called the impulse response function. Both are linear shift invariant (LSI) operators, which obey both the superposition principle

$$h0 (f0 + f1) = h0f0 + h0f1$$

and the shift invariance principle.

The simplest filter to implement is the moving average or box filter, which simply averages the pixel values in a KxK window. It is used as a pre-processing stage to edge extraction and interest point detection algorithms.

1.2.3. Non-linear filtering

Linear filters can perform a wide variety of image transformations; however, non linear filters can sometimes perform even better. One of these filters, the median filter, selects the median value form each pixel's neighbourhood, and is able to filter away such bad pixels. Other case is the bilateral filter, which simply rejects the pixels whose values differ too much from the central pixel, and the output pixel value depends on a weighted combination of neighbouring pixel values.

Other examples of neighbourhood operators include semi-global operator that computes distance transforms. It has many applications, including binary image alignment, feathering in image stitching and blending, and nearest point alignment.

Finally, another practical semi-global image operation is finding connected components, defined as regions of adjacent pixels that have the same value or label. Connected components are used in a variety of applications, such as finding letters inscanned documents or finding objects in images.

1.2.4. Image pyramids and wavelets

Neighbourhood operators can be cascaded to form image pyramids and wavelets, for analyzing images at a variety of resolutions and for accelerating certain operations. There are two possibilities for changing image resolution: interpolation and decimation.

In order to interpolate (or up sample) and image to a higher resolution, it is necessary to select some interpolation mask with which to convolve the image. On the other hand, there is decimation (or down sample), which is required to reduce the resolutions, where first the image convolves with a low-pass filter (to avoid aliasing) and then keep every sample.

A panorama is any wide-angle view or representation of a physical space, whether in painting, drawing, photography, film/video, or a three-dimensional model. A panoramic view is also proposed for multi-media, cross-scale applications to outline overview (from a distance) along and across repositories. This so called a cognitive panorama is a panoramic view over and a combination of cognitive spaces. Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. Commonly performed through the use of computer software, most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results. Some digital cameras can stitch their photos internally.

Image stitching can be performed using image pixels directly - correlation method; in frequency domain - fast Fourier transform method; using low level features such as edges and corners; using high level features such as parts of objects. Brown classifies image registration according to following criteria: type of feature space, type of search strategies and type of similarity measure. Image stitching originated in the photographic community, where more manually intensive methods based on surveyed ground control points or manually registered tie point have long been used to register aerial photos into large-scale photo-mosaics. One of the key advances in this community was the development of bundle adjustment algorithms, which could simultaneously solve for the locations of all of the camera positions, thus yielding globally consistent solutions. Another recurring problem in creating photo-mosaics is the elimination of visible seams, for which a variety of techniques have been developed over the years like the patch-based translational alignment (optical flow) technique. [10] Variants of this algorithm are used in almost all motion-compensated video compression schemes such as MPEG and H.263. Similar parametric motion estimation algorithms have found a wide variety of applications, including video summarization, video stabilization, and video compression. More sophisticated image registration algorithms have also been developed for medical imaging and remote sensing.

2. METHODOLOGY

Image stitching uses several key aspects and steps to implement and form a panorama. These steps are crucial to the working of the pipeline and provide a broad outlook of the techniques required to execute the workflow efficiently. The stitching pipeline is elaborated on in the following sections and a flowchart depicting the broad aspects of image stitching are visualized in Figure-1.

Three image stitching programs were written and executed to compare the efficiencies of the various methods entailed at each step of the stitching process and the workflow is established in the following sections. These three methods exploited various aspects of the image stitching pipeline.

After the image is acquired, if the image parameters are unknown, feature detection and extraction is carried out without Undistortion and cylindrical warping. In this case, the SIFT features are calculated from the acquired images and the images are translated based on the obtained homography. Following this, the images are blended using alpha or weighted blending.

If the image parameters, such as focal length and intrinsic parameters are known after image acquisition, the image first undergoes undistortion and warping. After that, two approaches are followed to compare efficiencies between the methods. The first approach involves the direct method in which the translation between the images is calculated without the need for features. This approach reduces the calculated time manifolds and aids in faster translation approximation.

After the translation between the images is calculated, feathering is performed to blend the images seamlessly and obtain the stitched image.

In the second approach, the feature based method is used wherein the SIFT features are calculated for the image sets. After that, the features are used to calculate the homography between the two images and determine the translation. After the homography is calculated, the images are stitched using a different blending technique called pyramid blending. This technique proves slightly more efficient than the previous technique due calculation of Laplacian and Gaussian pyramids of the images.

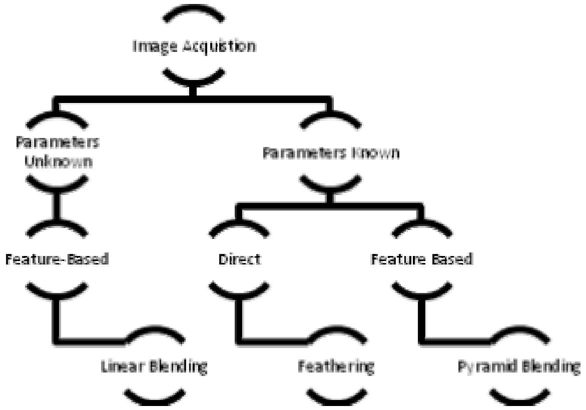


Figure-1. Workflow of methodology adopted.

2.1. Method-1

The image stitching software was implemented in Matlab with a main.m file using a series of function modules to complete the entire work flow. The workflow consists of four steps that execute in the following order.

2.1.1. Image acquisition

The images are acquired in one of two ways. Either the images are obtained from an existing set of images stored in a folder in the executed directory or the images are captured via the system's webcam for real-time execution of the image stitching program.

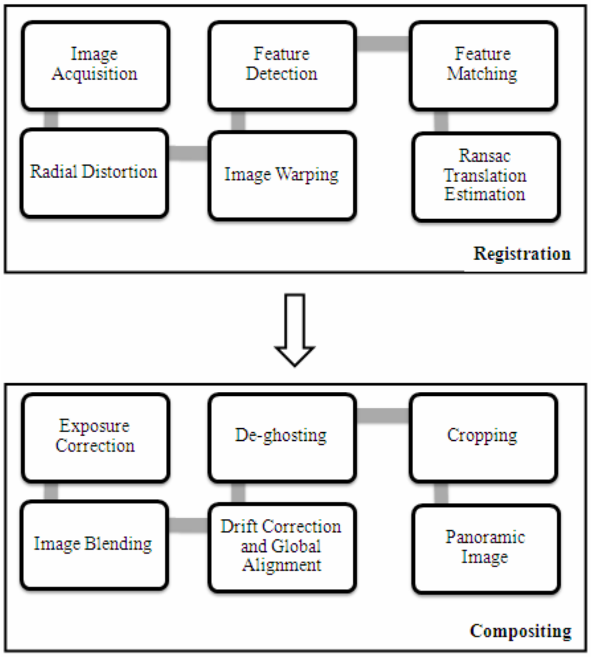


Figure-2. Flowchart describing stitching pipeline.

2.1.2. Feature detection and matching

Scale-invariant features (SIFT) are extracted from the acquired images. Each SIFT feature gives an output in the following form.

- Descriptor matrix: a K-by-128 matrix, where each row gives an invariant descriptor for one of the K keypoints. The descriptor is a vector of 128 values normalized to unit length.
- Location matrix: K-by-4 matrix, in which each row has the 4 values for keypoint location (row, column, scale, orientation). The orientation is in the range [-PI, PI] radians.

A MATLAB functional module sift was written to call the sift (for Linux) or the siftWin32.exe (for Windows) executable provided by Lowe, 2005 and extract all SIFT features for an input image object.

2.1.3. RANSAC translation estimation

Given the features of two scene-connected images, the matching features of the two images need to be found so a translation from one image to another can be estimated.

- **Lowe's matching**

Lowe's method is used for initial keypoint matching. Since the sizes of feature sets are typically large and different among images, it is infeasible to directly calculate the translation without knowing which in one set match which in the other set. Lowe's method is used to find a list of pairs for two images, each of which contains a feature in one image and its matching feature in the other image.

The Lowe's matching method is executed using the following concept. Let f_i be SIFT feature in image A and f_j is the SIFT feature in image B whose descriptor vector has the smallest Euclidean distance, $d_{smallest}$, from the descriptor vector of f_i. f_k is the SIFT feature in image B whose descriptor vector has the SECOND smallest Euclidean distance, $d_{2nd\ smallest}$, from the descriptor vector of f_i. Lowe's method states that if the following equation is satisfied, (f_i, f_j) forms a matching pair for images A and B:

$$d_{smallest}/d_{2nd\ smallest} <= \text{distance ratio}$$

Where, the distance ratio is a positive number smaller than 1. The smaller the distance ratio is, the fewer (and typically more accurate) matching pairs will be found. In the implementation of this project, we chose a distance ratio of 0.6. Additionally, calculating Euclidean distance is not efficient enough in Matlab, so the cos of dot product of the two descriptor vectors is calculated instead. The reasoning for doing so is that the ratio of angles (a cos of dot products of unit vectors) is a close approximation to the ratio of Euclidean distances for small angles. After Lowe's matching, the number of matching pairs between two images is typically below 100, which is much smaller than the total number of features per image.

- **RANSAC iteration**

The pairs given by Lowe's matching still cannot be used to estimate the translation. The reason is that there are still a fraction of outliers (false positives) that would pollute the estimation of translation. To eliminate the outliers, the RANSAC algorithm is used.

The process for RANSAC is explained in the next section. A basic outlook of the steps involved in carrying out the method is described as follows. A probability of $p = 0.5$ is chosen as that of encountering an in lier during a random pick. The probability of being success after RANSAC is set to be $P = 0.99$. For each iteration of the RANSAC, since we are only estimating 2D translation, only one sample pair is needed. Based on such setup, the number of iterations to run RANSAC is then $N = \log(1\text{-}P)/\log(1\text{-}p) = 7$. In each iteration, the initial estimated translation from image 2 to image 1 (T') is calculated based on the sample pick. Then the program goes through the rest of the matching pairs and collects inliers that satisfy the following equation:

$$|X'+T'-X|/2 < e \quad (1)$$

Where, X' is the coordinates of the feature in image 2 and X is the coordinates of the matching feature in image 1.e is a tolerance factor, and we chose 5 for this purpose. If the resulting number of inliers (including the random pick) is larger than that of the previous iteration, the set of these inliers is stored as a current set. After all iterations, the resulting current set is the final set of inliers.

The final translation, T, is estimated as the average translation among these inliers.

2.1.4. Image stitching and blending

The images are stitched in a pipelining manner. The first two images are stitched based on the estimated translation between them, producing an intermediate stitched image. Then the next image is stitched to the intermediate image, resulting in an updated intermediate image, so on so forth until the first image is stitched again to the end. When the two images are translated, a canvas is created. The dimensions of the canvas are calculated by creating masks of the two images and estimating the overlapping areas.

After creating the stitching "canvas", image A and image B are re-registered onto the canvas. The new row and column indices for each pixel in image A and image B are calculated.

Blending is needed to achieve a smooth transition within such intersecting area. In this project, linear blending is used along with a Gaussian filter on the mask to smoothen out the overlapping regions. The following formula is used to blend the two images together.

$$g(x) = (1-\alpha)\, f_0(x) + \alpha\, f_1(x) \quad (2)$$

After the blending is applied, the stitched image is obtained.

2.2. Method-2

The workflow consists of seven steps that execute in the following order: 1) the program acquires the input images from the folder and undistorts it. 2) Each undistorted image is then projected onto a cylindrical coordinate system. 3) SIFT features are extracted for each cylindrical image object. 4) Assuming the image objects are ordered in the way their scenes connect, the translation, T, to translate one image to its previous image's cylindrical coordinates is calculated. 5) The image objects are stitched and blended in a pipelining manner based on the translation calculated in the previous steps. 6) Vertical drift of the panorama is corrected using an affine transformation. 7) The image is cropped so the black edge areas resulting from the previous undistortion and cylindrical projection are removed, leaving only scene pixels. Each step is implemented in one or more functional modules.

2.2.1. Image acquisition and undistortion

The images are obtained from an existing set of images stored in a folder in the executed directory. Due to the nature of lens, the scene near the boundaries of an image can often be distorted and need to be corrected before panorama creation. In this project, the undistortion process is implemented as an inverse warping process. For each pixel (x', y') in the to-be-generated distortion-free image, the module calculates the pixel (x, y) of the same scene point in the original image and use its depth value(s) for (x', y'). The equations for calculating (x, y) are given below, where f is the camera's focal length in pixels,

$$r^2 = (x'/f)^2 + (y'/f)^2 \quad (3)$$

$$x = x' \times (1 + k1 \times r^2 + K2 \times r^4) \quad (4)$$

$$y = y' \times (1 + k1 \times r^2 + K2 \times r^4) \quad (5)$$

The (x, y) calculated using the above equations can be decimal numbers instead of integer pixels. In case this happens, the depth for (x', y') is calculated as the bilinear interpolation of the depths of the surrounding pixels of (x, y).

2.2.2. Cylindrical warping

The cylindrical reprojection was implemented as an inverse warping, too. The procedure is the same as that of the distortion removal, except that the following transformation equations are used instead.

$$x = f \times \tan(x'/f) \quad (6)$$

$$y = y' \times \mathrm{sqrt}(x^2 + f^2) \quad (7)$$

2.2.3. Feature extraction and matching

SIFT features are extracted from images using the same procedure described in Section 2.1.2.

2.2.4. RANSAC translation estimation

The translation is estimated using RANSAC iteration and Lowe’s method using the same procedure described in Section. 2.1.3.

2.2.5. Image stitching and blending

Image stitching is carried out based on the estimated translation between the obtained images. This produces an intermediate stitched image following which the next image is stitched to this intermediate image, resulting in an updated intermediate image, and so on so forth until the first image is stitched again to the end.

Assuming the translation from image B to image A is T (Δr, Δc) (in the form of row and column indices), so that (r_A, c_A) = (r_B, c_B) +T, the resulting size of the stitched image between A and B is calculated as:

$$\text{Total Rows} = \max(\text{Rows}_A, \text{Cols}_B + \Delta r) - \min(1, 1 + \Delta r) + 1$$

$$\text{total Columns} = \max(\text{Cols}_A, \text{Cols}_B + \Delta c) - \min(1, 1 + \Delta c) + 1$$

After creating the stitching "canvas", image A and image B are re-registered onto the canvas. The new row and column indices for each pixel in image A and image B are calculated as:

$$(r, c) = (r_A, c_A) + S$$

$$(r, c) = (r_B, c_B) + S + T$$

$$S = (1 - \min(1, 1 + \Delta r), 1 - \min(1, 1 + \Delta c)) \quad (8)$$

In the third equation, S is called the global shift. When stitching the first two images, the translation T is already known as the pair wise translation, but for successive stitching, the translations are cumulated. Assuming T_i and S_i are the translation and global shift in the i^{th} stitching and T_i^p is the pair wise translation from the next image to its previous image, the translation for the next stitching is calculated as:

$$T_{i+1} = T_i + T_i^p - S_i \qquad (9)$$

The above stitching algorithm can register some area of one image and some area of the other image to the same area in the final image.

Blending is done using the Laplacian pyramid blending algorithm. Laplacian pyramid is an algorithm using Gaussian to blend the image while keeping the significant feature in the meantime. It downsizes the image into different levels (sizes) with Gaussian. Later, it expands the Gaussian in to the lower lever and subtracts from the image in that lever to acquire the Laplacian image. After generating Laplacian pyramids for the overlap images A and B, the two images in different Laplacian levels are combined by combining partial images from each of them. Afterwards, the combined images are expanded from the top level to the next level (N-1) and add it to the original Laplacian image in the corresponding layer () to generate the latest Laplacian image in the corresponding layer. This step is repeated until reaching ground level and the final result will be the blending image.

2.2.6. Drift correction

After stitching, the program crops the left and the right areas of the resulting image so that the left edge and the right edge contain the centre column of pixels of the first image. A vertical affine transformation is applied onto this image to bring the left-edge pixel and the right-edge pixel with the same scene point to the same row. The vertical "drift" is calculated as the summation of all pairwise vertical translations.

2.2.7. Cropping

The program does auto cropping by removing all the rows in the image that contain at least one non-scene black pixel resulting from the undistortion and cylindrical projection steps. A naive algorithm will just detect for black pixels whose three color depths are all zeros. However, this might not work expectantly when there are actual black scene points. A more reliable way is to associate with each pixel a bit that indicating whether it is corresponding to a scene point and do cropping based on this information.

2.3. Method-3

The third approach is based on feature-less image stitching and translation. This approach uses the spectral properties of the image to compute the translation between the images following which the images are blended. The steps involved in this procedure are very straight forward and can be completed in the following four steps.

2.3.1. Image acquisition and undistortion

The images are acquired from a folder and undistorted using the same procedure described in Section. 3.2.1.

2.3.2. Cylindrical warping

After undistortion, the images are warped into cylindrical coordinates using the same procedure described in Section 2.2.2.

2.3.3. Frequency-based alignment

The Fast Fourier Transform is calculated for According to this property, also called as Fourier Shift Theorem, [19] if a certain function's origin is translated by certain units, then the translation appears in the phase of the Fourier transform. Suppose the two images I_1 and I_2 to be registered involve both translation and rotation with angle of rotation being 'θ' between them. When I_2 is rotated by θ, there will be only translation left between the images and the phase correlation with I_1 should give maximum peak. So by rotating I_2 by one degree each time and computing the correlation peak for that angle, we reach a stage where there is only translation left between the images, which are characterized by the highest peak for the phase correlation. That angle becomes the angle of rotation. Let Q (u, v) be the Phase correlation value of I_1' and I_2'rot, based on FI_1' and FI_2'rot.

$$Q\,(u,v)= \frac{FI_1'\,(u,v).\ FI_2'rot.^*(u,v)}{\left| FI_1'(u,v).\ FI_2'rot^*(u,v) \right|} \qquad (10)$$

After phase correlation is found between the images, the inverse Fourier transform is taken and a graph is computed to find the maximum peak of the graph. This gives the translation parameters between the images. Similarly, translation is calculated for all image sets and then stitching is performed.

2.3.4. Stitching and minimal blending

The next and last step in mosaicing is Image Stitching and Blending. At this stage, the reference image is overlaid on the source image by pasting its pixels on a canvas at the appropriate location using the transformation parameters obtained in the registration process. Pasting in the entire overlapped region in the final mosaic leads to a lot of redundancy and also degrades the quality of the mosaic. This algorithm is very efficient in stitching multiple images with large overlaps. This approach not only improves the efficiency of the stitching but the same time retains the quality of the mosaic closer to that of the input images. After this, the blending modifies the image gray levels in the vicinity of common boundary to obtain a smooth transition between images by removing the seams. Creating a blended image requires determining how pixels in an overlapping area should be presented. Performing blending in the entire overlapped region between the

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

images is not only time consuming but also leads to poor image quality. Further, when the overlap is very large, it could lead to false seams. Thus, for one image, within its intersecting area, the closer a pixel is away from the intersecting vertical boundary, the less percentage of its depth (intensity) values will be left in the stitched image.

3. IMPLEMENTATION AND SIMULATION

The algorithms involved in the process of stitching are explained in more detail here. These algorithms have been mentioned before in the image registration stage, and they are behind the processing of some specific tools employed for the stitching application: SIFT, RANSAC and Pyramid Blending, Feathering, Featureless Registration among others. These processes are specifically related to the control point detection and matching stage as well as the blending stage.

3.1. Algorithms

The following algorithms are explained in this section based on which they were implemented in MATLAB.

a) Scale Invariant Feature Transform (SIFT)
b) Random Sample Consensus (RANSAC)
c) Pyramid Blending
d) Frequency Based Alignment
e) Feathering

3.1.1. Scale invariant feature transform [1]

The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors". Each of these feature vectors is invariant to any scaling, rotation or translation of the image. This approach shares many features with neuron responses in primate vision. To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach:

a) Scale-space extrema detection

This stage of the filtering attempts to identify those locations and scales those are identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function. Further it has been shown under reasonable assumptions it must be based on the Gaussian function. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (11)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image.

Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (12)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

b) Keypoint localization

This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localised on an edge. This is achieved by calculating the Laplacian value for each keypoint found in stage-1. The location of extremum, z, is given by:

$$z = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}} \qquad (13)$$

If the function value at z is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation, it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the keypoint, the keypoint is rejected.

c) Orientation assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the keypoints scale to select the Gaussian smoothed image L, from above
- Compute gradient magnitude, m

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

- Compute orientation, θ
  $$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y)))$$
- Form an orientation histogram from gradient orientations of sample points
- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
- Some points will be assigned multiple orientations
- Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position

d) Keypoint descriptor

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centered on the keypoint. Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This result in a feature vector containing 128 elements.

3.1.2. Random sample consensus (RANSAC)

RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers, published by Robert B. Fisher. Normally, it is used at the feature matching stage, to determine good and bad keypoints connections between input images. It is a non-deterministic algorithm in the sense that it produces area sonable result only with a certain probability, with this probability increasing as more iteration are allowed.

The data consists of inliers of the predicted locations and outliers that do not fit the model. Usually, the outliers come from extreme values of noise or from erroneous measurements. RANSAC is the algorithm for robust fitting of models in the presence of many data outliers.

Assumptions

- Parameters x can be estimated from N data items.
- There are M data items in total.
- The probability of a randomly selected data item being part of a good model is p_g
- The probability that the algorithm will exit without finding a good fit if one exists is p_{fail} defined as $p_{fail} = (1 - p_g^N)^L$

Then, the algorithm is the following:

a) Selects N data items at random
b) Estimates parameter x
c) Finds how many data items of M fit the model with parameter vector x within a user given tolerance. Call this K.
d) If K is big enough, accept fit and exit with success.
e) Repeat 1 to 4 L times, defined as $\log(p_{fail})/\log(1 - p_g^N)$
f) Fail if it gets here.

The refined model is kept if its error is lower than the last saved model. The good keypoint matches remain to continue the process.

3.1.3. Pyramid blending

Laplacian pyramid is an algorithm using Gaussian to blend the image while keeping the significant feature in the meantime. The process is used to produce smooth blending between images. The steps involved in Laplacian Pyramid Blending are as follows:

- It downsizes the image into different levels (sizes) with Gaussian as shown in Figure-8.
- Later it expands the Gaussian in to the lower lever and subtracts from the image in that lever to acquire the Laplacian image.
- After generating Laplacian pyramids for the overlap images A and B, the two images in different Laplacian levels are combined by combining partial images from each of them, as shown in Figure-9.
- Afterward, the combined image from the top level is resized to the next level (N-1) and adds it to the original Laplacian image in the corresponding layer to generate the latest Laplacian image in the corresponding layer.
- This step is repeated until reaching ground level and the final result will be the blending image depicted in Figure-10.

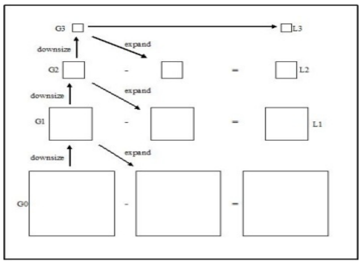


Figure-3 (a). First step to build Gaussian pyramids.

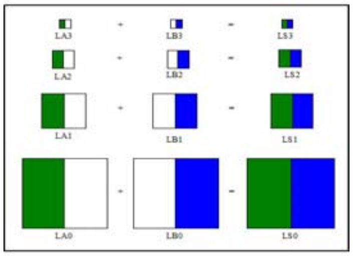


Figure-3 (b). Second step to generate Laplacian.

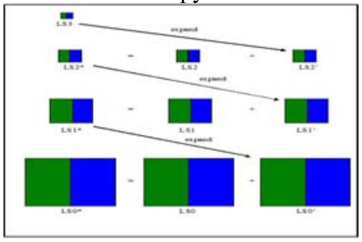


Figure-3(c). Final step to obtain blended version of two images.

3.1.4. Frequency based alignment

Frequency Based Alignment is used as a method to obtain translation between a set of images without using extraction of features from the images. This method is also known as a direct based method and consists of the following steps.

- The images are down sampled by 2 levels namely I_1 and I_2.
- I_2 is rotated by a certain step of degrees at each iteration namely I_2'rot.
- The Fourier Transform FI_1' and FI_2'rot of images I_1' and I_2'rot are calculated, respectively.
- The Phase correlation between the two calculated values is determined using the following formula.

$$Q\ (u,v) = \frac{FI_1'\ (u,v).\ FI_2'rot.*(u,v)}{\left| FI_1'(u,v).\ FI_2'rot*(u,v) \right|} \qquad (14)$$

- Following this, the inverse Fourier Transform q (x, y) is computed for the value, Q (u, v).
- The peak for the above transform is computed and stored in a vector at the particular position.
- The index of maximum peak is determined from the values stored in the vector in the previous step. This will give us the angle of rotation.
- The original image I_2 is rotated by the obtained angle and the new image obtained is I_2rot.
- The two images, I_1 and I_2rot are also phase correlated and the result is stored in P (u, v).
- The inverse Fourier transform p (x, y) of P (u, v) is determined and the peak is noted at position (tx, ty).
- The (tx,ty) become the translation parameters between the respective images.

3.1.5. Feathering

Feathering is a comparatively less complex blending method which aids in the blending of two images. When image pairs are combined, the seam is likely to be visible due to variance in pixel values and brightness. The feathering algorithm is implemented to blend images and reduce seam visibility.

x_{start} as the column where the image on the right begins and x_{end} as the column where the image on the left ends. The value of a pixel in column x, where x_{start}<= x <= x_{end}, is found using the following formulas.

$$w_{left} = 1 - (\ (\ x - x_{start}) / (x_{end} - x_{start})\ ) \qquad (15)$$

$$w_{right} = (\ (\ x - x_{start}) / (x_{end} - x_{start})\ ) \qquad (16)$$

$$x_{val} = x_{left}\ (\ w_{left}\ ) + x_{right}\ (\ w_{right}\ ) \qquad (17)$$

This relatively simple algorithm was found to be perfectly suitable for the images used; No seams could be identified through visual inspection of the panoramas produced.

3.2. Issues

The following algorithms are explained in this section based on which they were implemented in MATLAB.

a) Shearing/drift correction
b) Gain compensation and exposure correction
c) Gradient domain blending
d) De-ghosting

3.2.1. Drift correction

After stitching, the program crops the left and the right areas of the resulting image so that the left edge and the right edge contain the centre column of pixels of the first image. A vertical affine transformation is applied onto this image to bring the left-edge pixel and the right-edge pixel with the same scene point to the same row. The vertical "drift" is calculated as the summation of all pair wise vertical translations (including the translation from the first image to the last image).

The amount of vertical drift can be checked by added the first image to the end of the panorama and computing the vertical difference between the centres of the copies at either end of the panorama. The y coordinate of any point (x, y) in the mosaic is shifted as follows:

$$y' = y + ax$$

Finally, half of the first image is removed from the beginning and end of the panorama, so that the two edges line up perfectly to form a 360 degree view. The image is vertically cropped to remove any undefined pixels and the resulting panorama is output.

Figure-4. Affine drift due to movement of camera while acquisition.

3.2.2. Gradient domain blending

Sometimes, after the blending technique, the seams are still visible due to the exposure difference. Even though gain compensation is an effective method to get rid of the seam, there are still some portions of the seam that are visible in the image. Gradient Domain Blending is applied to such images to remove the seam as shown in Figure-13. The algorithm for the blending is as follows.

- Compute the derivatives of the input images dI_1/dx, dI_1/dy, dI_2/dx, dI_2/dy.
- Stitch the derivative images to form a field F = (F_x, F_y). F_x is obtained by stitchingdI_1/dx and dI_2/dx and F_y is obtained by stitching dI_1/dy and dI_2/dy.
- Find the mosaic image whose gradients are closest to F.

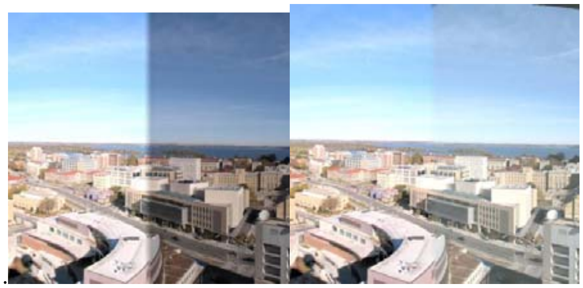


Figure-5. Artifacts due to exposure difference between images and effect of gain compensation.

3.2.3. Gain compensation and exposure correction

Since the image stitched has some artifacts in the form of exposure difference, gain compensation is done to remove these artifacts as shown in Figure-5. This can be done in the following steps.

- For each pair of overlapping images, the total difference for each color channel in the overlapping area is calculated.
- Then each difference is divided by the number of pixels in the overlapping area to get the average.
- After that, all the pixels are adjusted in the second image with these averaging differences to make its exposure match the first image.
- Due to accumulated adjustments, the last image in the sequence may have a very different exposure in comparison with the first image.

This causes a big jump in exposure when we connect the two ends of the panorama. To eliminate this effect, for the last pair of images, we only adjust the second image up to the centre of this image. In addition, instead of changing each pixel exactly the amount of averaging difference, we use a linear function such that there's a smooth transition at the centre of the second image. At the end, we crop the final panorama at the centre of the first image.

Figure-6. Gradient domain blending applied to remove seams.

3.2.4. De-ghosting

Sometimes when an image with a moving object is blended with other images, the moving object may become a ghost due to the blending process. In this process, a graph of the mismatching regions is created and the values from images that form a vertex cover in the graph are ignored. [24] Each image is weighed on its position so that it uses the pixels from images where the mismatched part is nearer to the center. Whether the mismatching pixels are completely subsumed in one particular image is also taken into account and its weight is boosted. To obtain chunks of mismatched columns rather than much small stretch of columns, the threshold is reduced every time one mismatched column occurs and it is increased again if the columns start matching.

Figure-7. Example of ghosting in the middle region.

3.3. Camera calibration

Before the implementation for the three aforementioned methods is implemented, two methods require calibration of the camera to determine the focal length and intrinsic parameters for warping and undistortion. The procedure for camera calibration is very straightforward and requires the camera calibration toolbox. Once the toolbox is obtained, the following steps are followed.

- A checkerboard pattern is create or obtained and pasted on a cardboard surface.
- The photos of this checkerboard are taken from several angles using the camera and then loaded onto the toolbox.
- Once these photos are obtained, the necessary steps regarding camera calibration are followed as per the given guide.

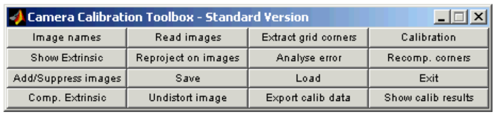


Figure-8. Camera calibration toolbox.

- The following parameters are obtained after following the steps given in the guide.

Focal Length: fc = [2566.27978 2568.64797] ± [10.44130 10.83100]
Principal point: cc = [1131.64444 864.82564] ± [18.95235 14.10695]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 deg
Distortion: kc = [0.02320 -0.09780 0.00036 -0.00314 0.00000] ± [0.03477 0.31232 0.00219 0.00280 0.00000]

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

Pixel error: err = [0.68068 1.01588].

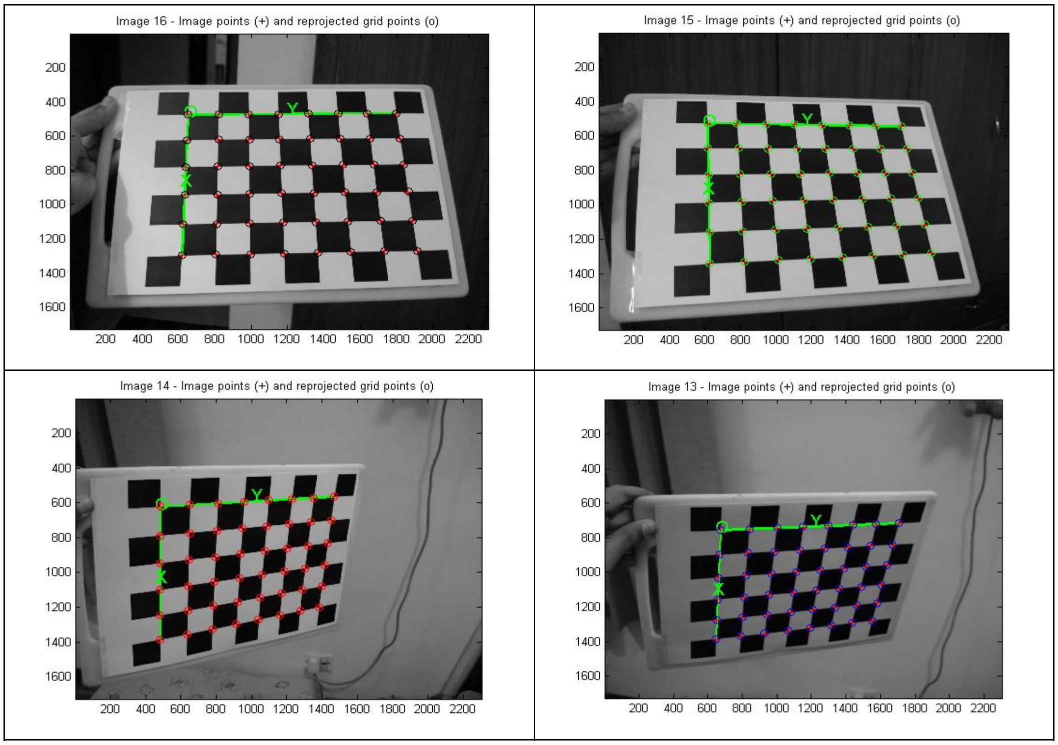


Figure-9. Calibrating camera using toolbox.

3.3.1. Sample output

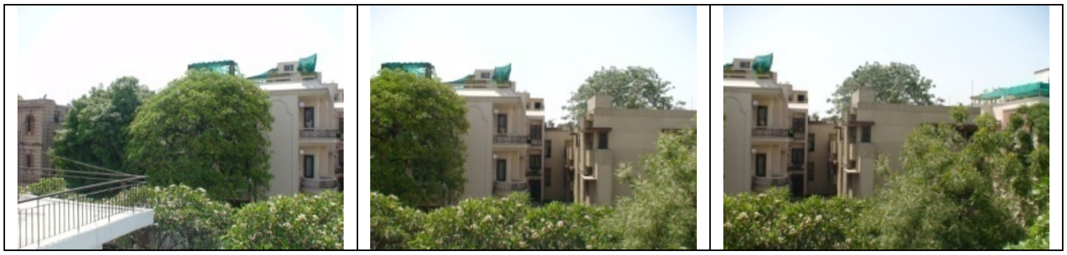


Figure-10. Input Images for method-1.

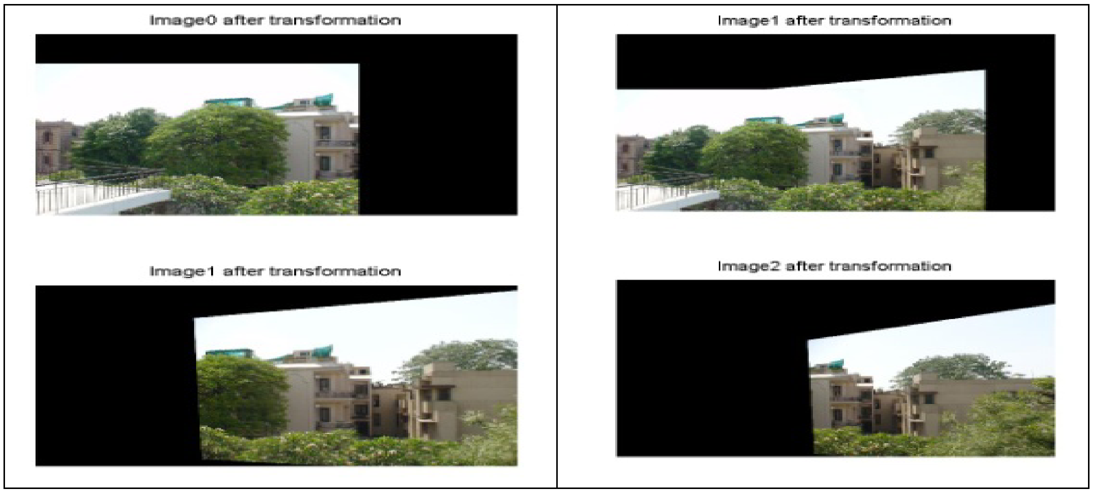


Image0 after transformation
Image1 after transformation
Image1 after transformation
Image2 after transformation

Figure-11. Image transformation between the pair of images after both iterations.

Stitched image after 1 iterations
Stitched image after 2 iterations

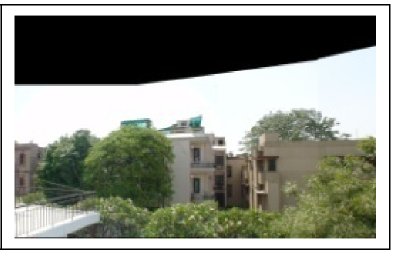


Figure-12(a) and (b). Stitched images after both iterations. **Figure-13.** Panoramic Image obtained from method-1.

3.4. Implementation of method-2

3.4.1. Sample output

Figure-14. Input Images for method-2.

Figure-15. Undistortion and warping of all images.

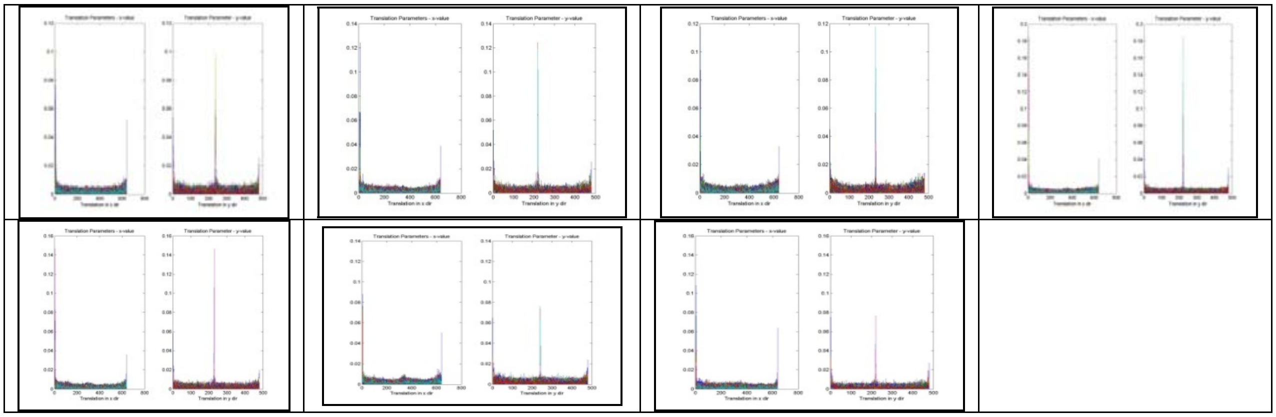


Figure-16. x-axis and y-axis translation between all images in one color channel.

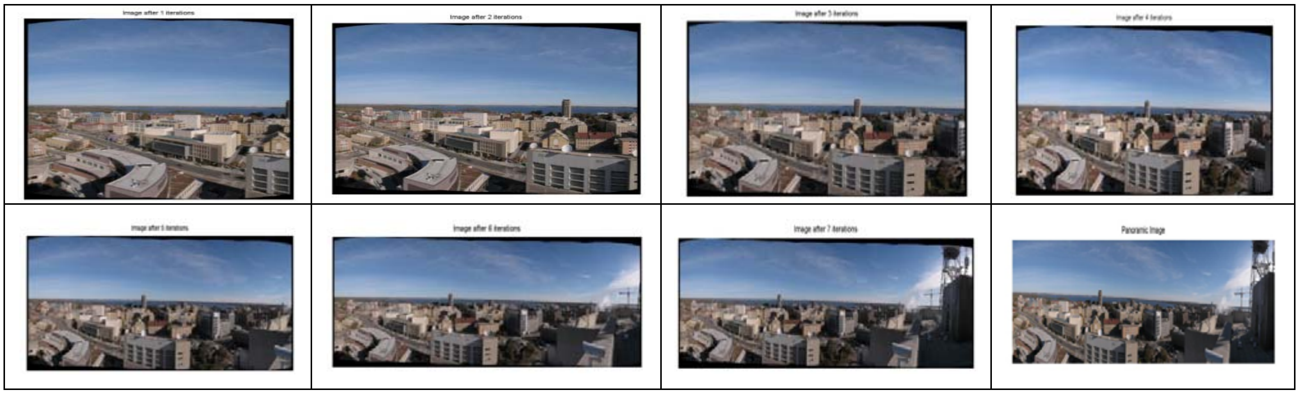
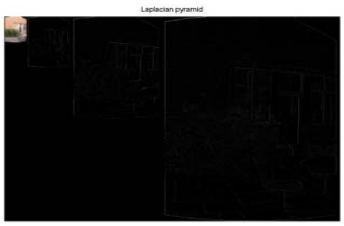


Figure-17(a). Stitched image after every iteration n. **Figure-17(b).** Panoramic Image obtained from method-2.

3.5. Implementation of method-3

3.5.1. Sample output

Figure-18(a) and (b). Input images for method-3. **Figure-18(c) and (d).** Undistorted and warped form of input images.



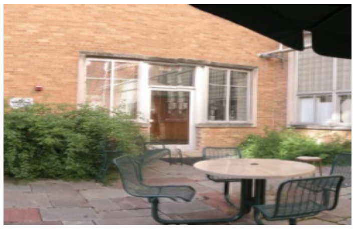


Figure-18(e). Gaussian image pyramids for 1st image. **Figure-18(f).** Laplacian pyramid for 1st image.

Figure-8(g) and (h). Overlapping region of pyramid (left) and blended image (right). **Figure-18(i).** Panoramic image obtained from method-3.

4. ANALYSIS OF OBTAINED OUTPUT

Once the output of the three methods is obtained, it is necessary to compute the efficiency for each of the three methods. For overview, the three methods are as follows:

a) The input is taken from a direct source such as the webcam or an input folder. The camera parameters are not calibrated and therefore, unknown. Thus, warping and undistortion is not performed on the input images. Following this, the SIFT features are computed for the input images. The location vectors are matched using the RANSAC algorithm which provides the homography in return. The homography helps us estimate the translation between the two images. The images are then stitched using alpha weighted blending.

b) The input is taken from a calibrated camera source which has known focal length and intrinsic parameters. The image is then undistorted and warped to remove any lens distortions. The Fourier Transform for the pair of image is calculated to compute the phase correlation. The peak of the inverse of this correlation gives the translation matrix at a particular angle of rotation. After this, the images are translated based on the obtained matrix and then the images are stitched together using feathering technique

c) The input is taken from a calibrated camera source which has known focal length and intrinsic parameters. The image is then undistorted and warped to remove any lens distortions and then the SIFT parameters are calculated for the two images. After calculating these parameters, RANSAC is used to identify inliers and select any five points which compute the homography. The homography matrix is used to calculate translation between the pair of images following which the images are blended using pyramid blending.

For efficiency purposes, the improvements in gain compensation, exposure correction and drift removal are ignored since they contribute to very less computation time as compared to the other process.

After executing the above three methods, the efficiency time of all three methods were obtained as follows:

Method-1
Calculation of SIFT: 5.3133s
Computation of homography using RANSAC: 0.66533s
Alpha weighted blending: 2.7568s

Method-2
Warping and Undistortion: 47.6152s
Calculation of phase parameters and translation: 2.843s
Feathering: 0.75034s

Method-3
Warping and Undistortion: 47.1866s
Calculation of SIFT: 4.2036s
Computation of homography using RANSAC: 1.5483s
Laplacian Pyramid Blending: 2.0478s

Since warping and undistortion are camera based and take excessive time, we assume that the calibrated images will be stored in the system and therefore, comparison is done on the remaining computation times.

These values can be better understood with the help of a table.

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

Table-1. Computation time for Method-1.

Method-1	1 image	2 images	10 images
Calculation of SIFT	2.6566	5.3133	26.566
Computation of homography	-	0.66533	3.32665
Alpha weighted blending	-	2.7568	13.784
Total Time (in sec)		8.73543	43.6766

Table-2. Computation time for Method-2.

Method-2	1 image	2 images	10 images
Phase Translation	-	2.843	14.215
Feathering	-	0.75034	3.7517
Total Time (in sec)		3.59334	17.9667

Table-3. Computation time for Method-3.

Method-3	1 image	2 images	10 images
Calculation of SIFT	2.1018	4.2036	21.018
Computation of homography	-	1.5483	7.7415
Laplacian Pyramid blending	-	2.0478	10.239
Total Time (in sec)		7.7997	38.9985

A few observations can be made based on the Table and computed values shown above.

- While the computation time is approximately the same for a pair of images for all three methods, as the number of images increases, the execution time increases at different rates.
- Method 2, i.e. frequency based alignment and feathering is computationally least expensive. However, the translation calculation based in this method is only rotation and illumination invariant. Any change in scale will not result in effective calculation of translation between the images.
- While both method 1 and 3 use SIFT features, the computation time in the latter is less expensive. This is due to undistortion and warping of the image which makes it easier to detect and extract features.
- Even though alpha weighted blending has fewer steps in implementation, Laplacian pyramid blending has lesser computation time. This is also because the image, in the case of Laplacian pyramid blending, is warped. This aids in faster Gaussian and Laplacian.
- Moreover, in alpha weighted blending, operations are done on every pixel in the overlapping area. Whereas in Laplacian pyramid blending, the image is treated as one entity and evaluated. This becomes less expensive for larger images.
- Another comparison of these methods can be depicted in the form of graphs as follows.

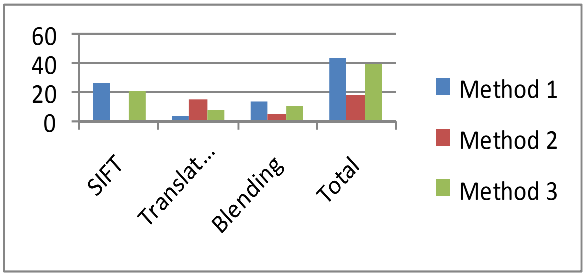


Figure-19. Graph comparing several steps of all three methods pyramids.

As it can be seen from the graph, method 2 is computationally least expensive if undistorting and cylindrical warping are not taken into consideration. Method 3 is the second quickest approach and should be adopted if the data is variant in scale. The first method can be used for real-time purposes when focal length and intrinsic parameters of the image are not known.

5. FUTURE WORK AND APPLICATIONS

Various methods to computer image stitching have been compared for efficiency and time constraints. While a lot of work has already been done in the field of panoramas, there are numerous applications which remain unexplored. The use of image stitching in these applications can prove to be beneficial with applications ranging from satellite imagery to video stitching.

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

5.1. Video stitching

A popular topic involving image stitching is video stitching. [21] While this problem is in many ways a straightforward generalization of multiple-image stitching, the potential presence of large amounts of independent motion, camera zoom, and the desire to visualize dynamicevents impose additional challenges. For example, moving foreground objects can often be removed using median filtering. Alternatively, foreground objects can be extracted into a separate layer and later composited back into the stitched panoramas, sometimes as multiple instances to give the impressions of a "Chronophotograph" and sometimes as video overlays. Videos can also be used to create animated panoramic video textures in which different portions of a panoramic scene are animated with independently moving video loops

5.2. Satellite imaging

Image stitching can be used to create panoramas of images captured from the satellite. Many a times, satellites capture pictures of a wide area to study characteristics and conditions of a certain region. Image stitching can be used in such scenarios to create image mosaics which give wholesome views of a region for analysis and study. They can also be used in strategic locations to map large areas of unknown or unfamiliar territories.

5.3. Medical imaging

Medical imaging is another crucial area where image stitching is vital. Micro imaging is performed on certain regions of the body to examine malign conditions. For example, tumorous sections are examined using micro imaging tools to map areas of the body which are infected. Image stitching is vital at such a juncture to provide overlapping images and create a mosaic of the affected and surrounding area at high resolution.

5.4. Steganography

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. For hiding secret information in images, there exists a large variety of steganographic techniques some are more complex than others and all of them have respective strong and weak points. Image stitching and panoramas, in general, can aid in this process by increasing the carrying capacity manifold. Since panoramas are formed by merging numerous images together, the carrying capacity of the carrier, i.e. the panorama increases to a very high level such that data can be encrypted in a more secure manner and have higher quantity at the same time.

5.5. 3D objects

Image stitching can also be important for analyzing surface properties and textures of 3 dimensional objects. Images of an object can be taken from several aspects and stitched together to provide a comprehensive study of the characteristics and features of the object. Moreover, image stitching can also aid in reconstruction of 3D objects whose images have been acquired elsewhere. This forms an important application for computer graphics problems.

5.6. Object recognition

A lot of times, for image and object recognition purposes, one image does not have all components of an object required to be identified. Thus, if images of a scene are stitched together, it aids in more efficient and effective registration of objects. This also leads to a more robust performance due to more visible characteristics.

6. CONCLUSIONS

Image stitching is a vast field with numerous applications. While the concept may be old, new and innovative applications such as the ones mentioned above, make it an enticing field. Thus, it is crucial to understand the underlying principles behind image stitching and the fundamentals required to form image mosaics and panoramas. In this paper, several algorithms and processes involved in image stitching were studied and executed. Less related topics such as camera calibration and image distortion and warping were also studied to enhance implementation of the process. Three primary methods of image stitching were understood, implemented and analyzed. The first method aimed at acquiring images directly without any need for camera calibration. The other two methods required certain camera parameters but differed in terms of translation estimation and blending techniques. All three methods were analyzed for efficiency, computational time and effectiveness. Several issue such as gain compensation, exposure difference, ghosting and affine drift were also dealt with. A comprehensive analysis of each step of the procedures was done to understand the efficiency of each process. The stitching pipeline was successfully implemented in MATLAB and executed on numerous examples to show its efficacy. Finally, several applications and future prospects of image stitching were discussed with regard to numerous fields of application.

REFERENCES

[1] M. Brown and D. Lowe. 2007. Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision.

[2] A Levin, AZomet, S Peleg and Y Weiss. 2004. Seamless image stitching in the gradient domain. Computer Vision ECCV.

[3] Richard Szeliski. 2006. Image Alignment and Stitching. Foundations and Trends in Computer Graphics and Vision.

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

[4] Dahl A.L., Aanæs H. and Pedersen K.S. 2011. Finding the Best Feature Detector-Descriptor Combination. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT).

[5] Alahi A., Ortiz R. and Vandergheynst P. 2012. FREAK: Fast Retina Keypoint. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Tinne Tuytelaars and Krystian Mikolajczyk. Local Invariant Feature Detectors: A Survey Foundations and Trends in Computer Graphics and Vision.

[7] Rublee E., Rabaud V., Konolige K. and Bradski G. 2011. ORB: An efficient alternative to SIFT or SURF. IEEE International Conference on Computer Vision (ICCV).

[8] Yingen Xiong and Kari Pulli. 2009. Sequential Image Stitching for Mobile Panoramas. IEEE 7th International Conference on Information, Communications and Signal Processing.

[9] Piotr Ostiak. 2009. Implementation of HDR panorama stitching algorithm. Central European Seminar on Computer Graphics.

[10] Rebiere N, Auclair-Fortier M.-F and Deschenes F. 2008. Image mosaicing using local optical flow registration. 19th International Conference on Pattern Recognition. ICPR.

[11] Zomet A, Levin A, Peleg S and Weiss, Y. Seamless image stitching by minimizing false edges. IEEE Transactions on Image Processing.

[12] CY Chen and R Klette. 1999. Image stitching-Comparisons and new techniques. Computer Analysis of Images and Patterns.

[13] A. Baumberg. 2002. Blending images for texturing 3D models. In BMVC.

[14] F. Bernardini, I. Martin and H. Rushmeier. 2001. High quality texture reconstruction from multiple scans. IEEE Trans. on Visualization and Computer Graphics. 7(4).

[15] Y. Boykov and V. Kolmogorov. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on PAMI. 26(9).

[16] Y. Boykov, O. Veksler and R. Zabih. 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. On PAMI. 23(11).

[17] P. J. Burt and E. H. Adelson. 1983. A multiresolution spline with application to image mosaics. ACM Trans. On Graphics. 2(4).

[18] P. Labatut, J.-P. Pons and R. Keriven. 2007. Efficient multiview reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In ICCV.

[19] B. Reddy and B. Chatterji. 1996. An FFT-based Technique for Translation, Rotation and Scale-invariant Image Registration. IEEE Trans. on Image Processing. 5(8): 1266-1271.

[20] Q. Chen, M. Defrise and F. Deconinck. Symmetric phase-only matched filtering of Fourier-Mellin transforms for Image Registration and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 16(12): 1156-1168.

[21] R. Szeliski. 1996. Video mosaics for virtual environments. IEEE Computer Graphics and Automation. pp. 22-30.

[22] R. Szeliski and H. Y. Shum. Creating Full View Panoramic Image Mosaics and Environment Maps, Proc. SIGGRAPH'97. pp. 251-258.

[23] J. Davis. Mosaics of Scenes with Moving Objects. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98). pp. 354-360.

[24] M. Uyttendaele, A. Eden and R. Szeliski. Eliminating Ghosting and Exposure Artifacts in Image Mosaics.

[25] Assaf Zomet, Anat Levin, ShmuelPeleg and Yair Weiss. 2006. Seamless Image Stitching by Minimizing False Edges. IEEE transactions on image processing.

[26] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt and O. J. M. 1984. Pyramid method in image processing, RCA Eng. 29(6): 33-41.

[27] S. Peleg. 1981. Elimination of seams from photomosaics, Comput. Graph. Image Process. 16(1): 90-94.

[28] R. Fattal, D. Lischinski and M. Werman. 2002. Gradient domain high dynamic range compression, in Proc. SIGGRAPH. pp. 249-356, July.