ANALYSIS OF SIFT AND SURF ALGORITHMS FOR IMAGE MOSAICING ON EMBEDDED PLATFORM

By

OOI CHONG WEI

A Dissertation submitted for partial fulfilment of the requirement for the degree of Master of Science (Electronic Systems Design Engineering)

August 2015

Acknowledgement

During the period of doing the project, some people have helped and guided me. I would like to express my appreciation and thank to all of them. Firstly, I would like to thank Prof. Dr. Mohd Rizal bin Arshad who provides opportunity for me to work on my project under his supervision. Besides, thanks to him becoming my supervisor, providing his sincere comment and helpful advice to me. Thanks again for giving suggestion to my project. In addition, I appreciate his help to introduce one of his PHD students, Ms Herdawatie Abdul Kadir for providing some idea to me.

I would like to take opportunity to thank Ms Herdawatie. She had guided me when I faced some difficulties. Also, she gave me some useful information and comment on my project. Thanks for spending some of her precious time to meet me for project discussion. I would like to thank Mr Muhammad Faiz Abu Bakar who lends me some boards to work on my project.

I am grateful and thanks to all my friends who had helped me indirectly, providing moral support to me. Last but not least, I wish to acknowledge my parents and family members for their encouragement, supporting, caring when I am working on my project.

Table of Contents **Page**

List of Tables **Page**

List of Figures **Page**

List of Abbreviations

DLT	Direct Linear Transform
FAST	Features from Accelerated Segment Test
FLANN	Fast Library for Approximate Nearest Neighbors
JPEG	Joint Photographic Experts Group
LM	Levenberg Marquardt
OS	Operating System
RAM	Random Access Memory
RANSAC	Random Sample Consensus
SIFT	Scale Invariant Feature Transform
SSD	Sum of Squared Difference
SURF	Speeded-Up Robust Features
SUSAN	Smallest Univalue Segment Assimilating Nucleus

List of Symbols

σ	Scale space
λ	Eigenvalue
$\theta(x, y)$	Direction of image in coordinate (x, y)
d	Euclidean distance
H	Homography matrix
$H(x, y)$	Grayscale image
$m(x, y)$	Gradient of image
w	Window at position
x	Coordinate of image pixel in x axis
y	Coordinate of image pixel in y axis

Analisis Antara Algoritma SIFT dan SURF untuk Pencantuman Imej Dalam Platform Terbenam

Abstrak

Mozek atau cantuman imej ialah satu bidang penyelidikan yang aktif dalam penglihatan komputer. Proses cantuman imej ialah proses untuk mencantum beberapa imej yang mengandungi sebahagian paparan yang sama menjadi satu imej yang besar. Hasilan cantuman semua imej input dipanggil sebagai imej panorama. Teknik cantuman imej dikelaskan kepada dua jenis. Teknik pertama ialah teknik langsung dan teknik kedua dikenali sebagai teknik berasaskan ciri-ciri kandungan. Kebaikan teknik berasaskan ciri-ciri kandungan adalah dalam bentuk ciri keteguhan dan kelajuan. Imej panorama akan dicipta dengan lebih cepat dan mengandungi kenaikan kualiti hasilan. Dalam penyelidikan ini, sistem imej mozek dalam papan litar secara masa nyata berasaskan teknik ciri SURF telah dicadangkan. Prestasi antara SURF dan SIFT telah dibandingkan. Untuk memperolehi mata padanan antara imej, Flann Based Matcher teleh digunakan. Seterusnya, anggaran 'homography' telah dianggarkan dengan menggunakan algoritma RANSAC. Transformasi perspektif dipakai untuk mendapatkan transformasi bagi memetakan sisi empat berbentuk dua dimensi ke atas sisi empat lain. Akhir sekali, imej akan bergabung dengan imej lain menjadi satu gamber imej. Hasil eksperimen menggambarkan SURF dan SIFT adalah algoritma yang teguh dalam pencarian kunci utama yang stabil. Teknik-teknik ini tidak akan diganggu oleh perubahan skala dan putaran.Teknik SURF mempunyai prestasi yang lebih baik dalam ciri kelajuan. Pelaksanaan dan experimen telah dijalankan dalam papan litar bercetak Raspberry Pi yang mempunyai 256MB RAM dan pemproses 700MHz.

Analysis of SIFT and SURF Algorithms for Image Mosaicing on Embedded Platform

Abstract

In computer vision, image mosaicing or stitching is a common active research area. Image stitching process is a process of compositing images which contain similar scene into a larger image. The union of these input images is called panoramic image. Image stitching techniques are classified into two types. First technique is direct technique whereas another is known as feature-based technique. Significant pros of feature-based method are in terms of robustness and speed. As a result, panoramic image is created faster and contains quality improved. In this paper, a real time on board image mosaicing system based on SURF feature based techniques is proposed. Performance comparison between SURF and SIFT is made. To obtain matching point between images, Flann Based Matcher is used. Next homography estimation is performed by using RANSAC algorithm. Perspective transform is applied to obtain a transformation for mapping a two dimensional quadrilateral into another. Lastly, images are warped and composited into single scene. Experimental results shows that SURF and SIFT are robust algorithm performing stable key point detection. These techniques are invariant to scale and rotation. SURF technique has better performance with respect to speed. Implementation and experimental are done in Raspberry Pi board with built-in 512MB RAM and 700MHz processor.

CHAPTER 1

INTRODUCTION

1.1 Overview

Camera angle limits size of image captured. Due to this limitation, several functionality camera lenses (such as Canon EF-S 10-22mm f/3.5-4.5 USM lens with wide angle focal length) have been introduced in market. This improves wider angle of scene that can be captured in single scene but still have limitation in image size. Part of the scene or information might be left out since it is not able to be loaded and contained in single image. Hence panoramic images are introduced to composite images into single scene.

The aim of this project is to design and develop an embedded system to perform on board image stitching. The system design can perform basic camera functions such as capture images, store images, performs image processing and stitching. Multiple images of a scene will be constructed into a composite image for larger field of view. The result can allow users to view surrounding scene with real images composited into single scene.

Advantages of this development are users able to perform image processing at lower cost embedded board rather than performing in computer and low power consumption. Besides, image capturing and processing are performed in real time to provide result on sight. This development can be implemented in robot to capture surrounding image in wider view.

Limitation of the development is in terms of processing speed and storage. Due to the board limitation, the image captured is in low pixels (600 x 800). Without external RAM supported, the speed is slower as compared to computer.

1.2 Research Motivation

Proposed on board image stitching algorithm is developed to capture image and create panorama by stitching images together. The algorithm can detect features in image, extract descriptor for feature point matching, estimate homography and composite images into a scene. Therefore, this research will create an algorithm that can help to implement image panoramic feature in robots to view surrounding area and track targets. The users can use putty software to remote accessing the board through internet and perform tasks or operation required. The results and displays can be shown on monitor. In overall, this will ease users to capture and store larger view of images.

1.3 Problem Statement

Larger scene of image can be captured as the camera angle is wider whereas more information of the surrounding scene can be seen in single image. The images captured using standard camera has limitation in terms of image size and need to be loaded into computer to perform image processing and mosaicing for images combination. By developing real time on board image processing and mosaicing in embedded platform can produce panoramic image on the spot.

1.4 Research Objective

The objectives of this research are to develop algorithms for embedded board to perform basic camera functions such as capture, store and process images. Images are retrieved to perform image processing and stitching to create panoramic image. Besides, this research will also investigate feature detection techniques. The performance of these techniques will be evaluated. Process from raw image registration, image composite until panoramic image creation will be studied.

The sub objectives for this research listed in below:

- To study, investigate and evaluate performances of different feature detection algorithm.
- To embed feature detection and image mosaicing algorithm into embedded board for real time creating panoramic image.

1.5 Scope of Research

Proposed embedded system will be designed and developed for Linux application only whereas programming language used to develop algorithm are C++ and C programming. OpenCV library will be used to perform image processing. All the commands required will be executed through command terminal in Linux operating system. The algorithm will be tested in Raspberry Pi board with standard USB web camera connected. The captured image will be stored in Joint Photographic Experts Group (JPEG) standard format. Operating system of Raspberry Pi and memories will be stored in SD card. No external power usage except from 5V power supply to Raspberry Pi board. USB web camera power

consumption required will be supplied through USB port of the board. Due to limitation of memory and processing speed of the board, the images will be captured in lower pixels (600 x 800).

1.6 Thesis Organization

This thesis is written in five chapters. Chapter 1 is the overall concept of the research project. The chapter starts with overview, followed by motivation of the project. Objectives and scope of research are stated in this chapter.

Chapter 2 is literature review. In this chapter, overview on image mosaic is introduced upfront. Then different types of feature detection, extraction and matching algorithm are discussed. Homography estimation and image blending are reviewed in this section. Besides, usage and review on OpenCV are presented. Lastly related previous and current works are summarized.

Chapter 3 outlines methodology of this project. Design and implementation of on board image stitching are introduced. The details on design flow, hardware and software required are explained. In addition, algorithm used for image stitching is discussed.

Chapter 4 discusses and evaluates project outputs and result findings. Data of feature detection on benchmarking test images is collected and analyzed. Times taken in feature detection processing and method performances are available in this section. Besides, results and performance of descriptor matching are shown. Project output and result details are described and discussed in final sub section.

Chapter 5 summarizes the research project and provide conclusion. The contribution and findings of the project are highlighted. Recommendation for future work is provided in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Cameras have limited angle view to capture the scene of surrounding area in one shot. Today digital camera facilitated with panorama feature is available in market. However, quality of result is degraded due to blurring and ghosting impact. Camera movement speed and speed specification of the camera are vital criteria to create complete panorama image (Mahesh & Subramanyam, 2012).

Image mosaicing is common active research in image processing related computer vision. It is to construct multiple image of a scene to create a composited image for larger field of view. The image result can allow users to view surrounding scene with real images that composited into single scene. Three steps to construct image mosaicing system are input multiple images, perform image registration (Zitova B. & Flusser J., 2003) and produce mosaic image (P.Mittal & Y.P Singh, 2013).

Various methods are discovered and adopted for image mosaicing and known as image registration techniques. These techniques can be classified into feature based and featureless methods. Featureless or direct method is more useful to mosaic large overlapping area with small translations or rotations. Feature based method commonly used in mosaicing small overlapping area. This method will provide better result in terms of accuracy.

Assumption made for feature based methods is the correspondences between images are exist and image registration is done by utilizing these correspondences to search transforms. Major difficulty for these methods is the image feature acquisition and tracking. These are due to occurrence of noise and occlusion.

For featureless methods, transforms required in image registration are discovered by minimizing sum of squared difference (SSD) function. This will involve some of the formula parameters but do not rely on correspondences. Therefore, no issues related to feature acquisition and tracking. Major concern for these methods is the requirement for small changes occurrence (translation, rotation, etc.) from one image to another. Optimal solution is no guarantee to be achieved as the parameters initialization and estimation process might lead to different results. Image mosaic system basically is illustrated in flow chart below (P.Mittal & Y.P. Singh, 2013):

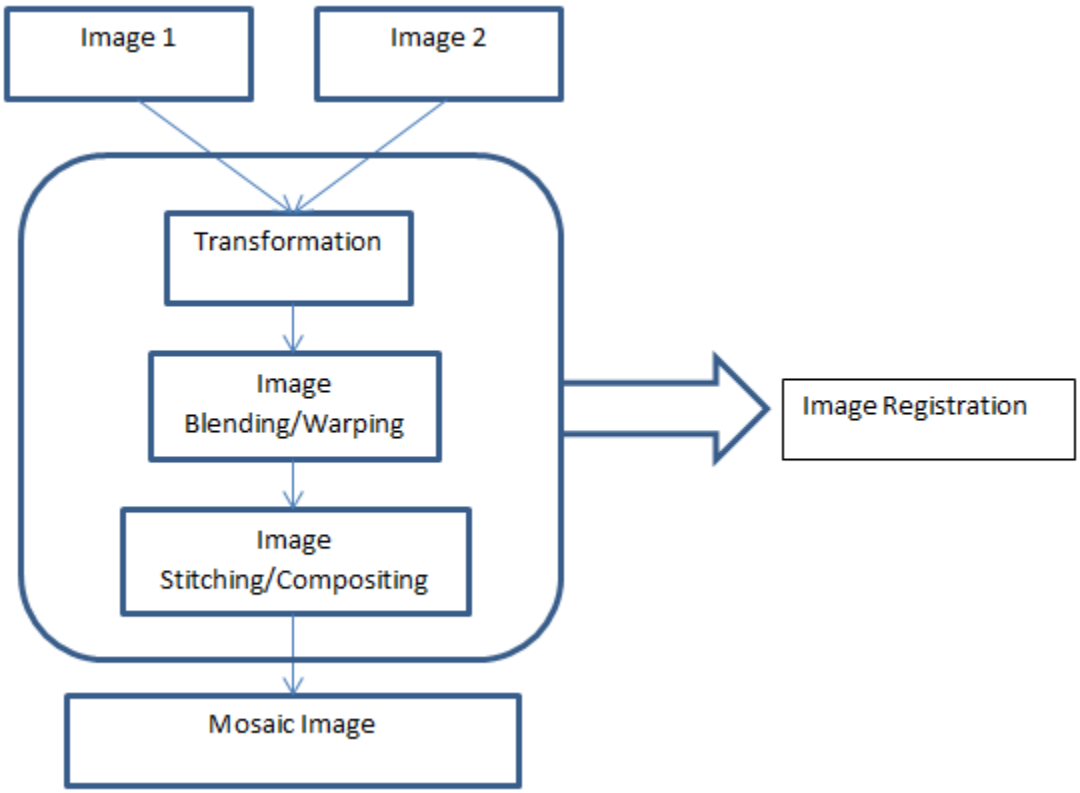

Figure 2.1 Flow Chart of Image Mosaic System (P.Mittal & Y.P. Singh, 2013)

Image mosaic algorithm integrated in embedded board enable on board image mosaicing to be performed in real time. With images that contains common interception can be stitched together to provide a larger view.

Features are characteristics of the matching points that can be referred in the scene. The characteristics are unique and can easily be recognized. The importance of these features is to find matching characters among images. With this detection, information of images can be extracted. Types of common image features are edges, corners and blobs.

2.2 Feature Detection Algorithm

2.2.1 Harris Corner

One of the well-known algorithms in corner detection is Harris corner detector. Early attempt of finding these corners was carried out and introduced by Chris Harris and Mike Stephens in 1988(Harris & Stephens, 1988). Corner is one of the feature points of interest or intersection of two edges. The direction change of two edges is represented in corner also. Shifting the window will cause significant change in appearance. To recognize corner, the variation must be determined (Yang et al., 2013).

In short, corner detection actually is summarized into two categories which are based on image edge and image grayscale. For image edge based algorithm, edge is detected and mutation point at this edge is recognized as corner. Algorithm based on image grayscale needs to calculate curvature and gradient whereas gradient maxima is considered as corner (Yang et al., 2013).

By taking grayscale image H into consideration, variation of intensity is calculated using Equation (2.1):

$$E(a,b) = \sum_{x,y} w\,(x,y)[H(x+a,y+b) - H(x,y)]^2 \qquad (2.1)$$

where $w(x,y)$ is window at position (x,y), $H(x,y)$ is intensity at position (x,y) and $H(x+a,y+b)$ is the intensity at new position window $(x+a,y+b)$ (Yang et al., 2013).

In order to find out windows with large intensity variation, Equation (2.1) has to be maximized:

For distinctive patches, $[H(x+a,y+b) - H(x,y)]^2$ is larger and $E(a,b)$ will be larger.

Applying Taylor Series on $H(x+a, y+b)$:

$$H(x+a,y+b) = H(x,y) + aH_x(x,y) + bH_y(x,y) + \frac{1}{2!}\left[a^2H_{xx}(x,y) + abH_{xy}(x,y) + b^2H_{yy}(x,y)\right] + \frac{1}{3!}\left[a^3H_{xxx}(x,y) + a^2bH_{xxy}(x,y) + ab^2H_{xyy}(x,y) + b^3H_{yyy}(x,y)\right] + \cdots (Higher\ order\ terms) \quad (2.2a)$$

Hence, first order approximation on $H(x+a, y+b)$ is shown in Equation (2.2b).

$$H(x+a,y+b) \approx H(x,y) + aH_x(x,y) + bH_y(x,y) \quad (2.2b)$$

Applying Taylor Series Expansion on $E(a,b)$:

$$E(a,b) \approx \sum_{x,y}[H(x,y) + (aH_x + bH_y) - H(x,y)]^2 \quad (2.3)$$

$$E(a,b) \approx \sum_{x,y} a^2H_x^2 + 2abH_xH_y + b^2H_y^2 \quad (2.4)$$

The summarize equation can be illustrated in matrix form as shown in Equation (2.5) and (2.7) with assumption in Equation (2.6):

$$E(a,b) \approx [a\ b]\sum_{x,y} w(x,y)\begin{bmatrix} H_x^2 & H_xH_y \\ H_xH_y & H_y^2 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} \quad (2.5)$$

Assuming $\sum_{x,y} w(x,y)\begin{bmatrix} H_x^2 & H_xH_y \\ H_xH_y & H_y^2 \end{bmatrix} = N$ (2.6)

Hence $E(a,b) \approx [a\ b]N\begin{bmatrix} a \\ b \end{bmatrix}$ (2.7)

Equation (2.8) is used to determine whether corner is possibly established (Ryu, 2011; Zhu & Yang, 2011; Liang et al., 2014):

$$R = \det(N) - k\left(trace(N)\right)^2 \quad (2.8)$$

where $\det(N) = \lambda_1\lambda_2$, $trace(N) = \lambda_1 + \lambda_2$ and k is empirically determined constant. If R greater than a defined value, it is considered as corner (Juan at el., 2011).

λ_1 and λ_2 are eigenvalues of the matrix N in which N represents the intensity of local surrounding area. Cases that required to be considered in Harris corner detection (Lei. Y., 2011; Mahesh & Subramanyam, 2012) are

i. If both λ are small, auto correlation function will be flat and windowed image region almost in constant intensity. This is flat indicator.
ii. If gaps between both λ are large (λ_1 is much larger than λ_2 or vice versa), auto correlation function for this is in ridge shaped, shift in single direction causing little changes in matrix N but significant change in orthogonal direction. This is an edge indicator.
iii. If both λ are big, auto correlation function is in sharp peak state, shifts in any direction will cause significant increment. This is a corner indicator.

Summary for all the cases is shown in Figure 2.2.

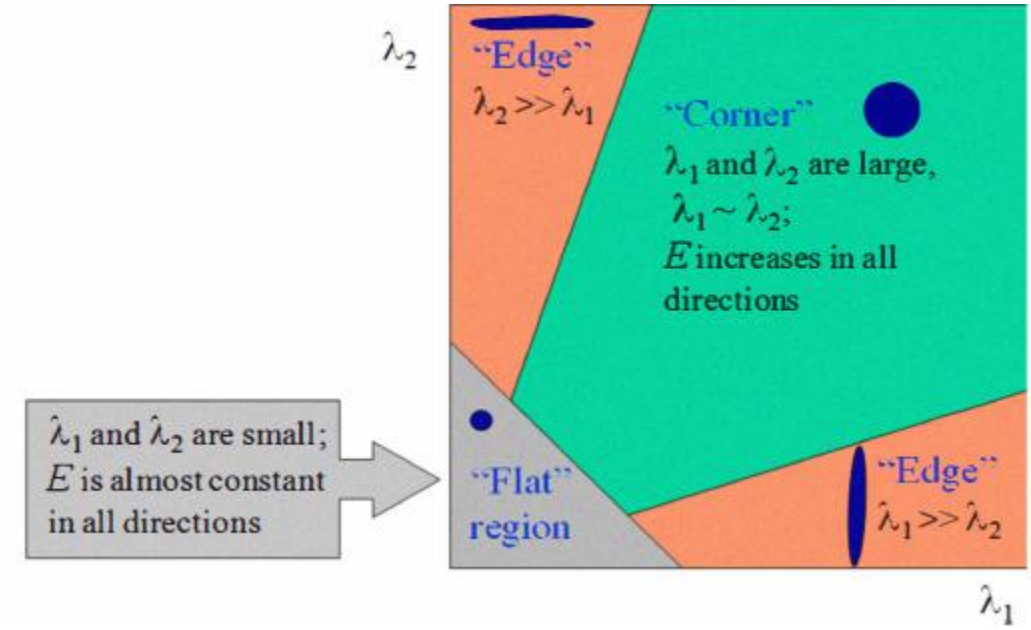

Figure 2.2 Indicator definitions in Harris (Mahesh & Subramanyam, 2012).

2.2.2 SIFT Feature Detection

Scale Invariant Feature Transform, SIFT extract unique features such as affine transformation, rotation, scale invariant and noise immunity characteristics. SIFT is also a feature detection algorithm that able to detect feature in images and produce key point descriptors (Wang et al., 2009). The two main function of key point descriptors are used as input to Nearest Neighbor Search (NNS) and produce closely matching key point descriptor.

For the initial stage of key point detection is to determine coordinates and scales that can be allocate repeatedly beyond different views of same object. By searching the stable feature in image across all possible scales, locations detection that invariant to scale change of images can be identified using scale space. The possible scale space kernel is known as Gaussian function.

The scale space that produced from convolution of variable scale Gaussian, Equation (2.10) with input image is defined in Equation (2.9) (Huang et al., 2012):

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.9}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)}/2\sigma^2 \tag{2.10}$$

Where $L(x, y, \sigma)$ is scale space image function,$G(x, y, \sigma)$ is variable (scale Gaussian variable) and $I(x, y)$ is input image.

Scale space extrema is used in difference of Gaussian function to detect stable key point location in Equation (2.11)

$$\begin{aligned} D(x, y, \sigma) &= \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{2.11}$$

To detect local maxima and minima of $D(x, y, \sigma)$, each point is compared with its eight neighbor points in current image and nine neighbor points of scale image above and below as illustrated in Figure 2.3 (Lei Yang et al., 2011).

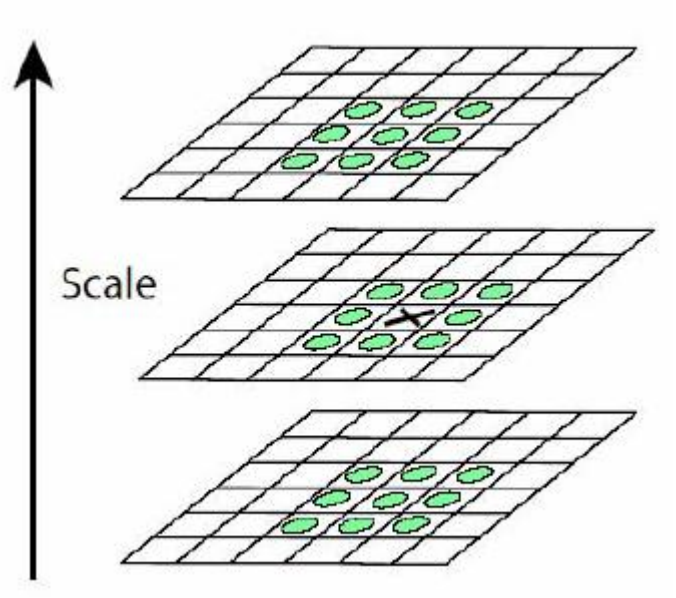

Figure 2.3 Point comparison with nearest neighbor (Mahesh & Subramanyam, 2012).

The point will only be chosen if it is larger or smaller as compared with all the neighbors. Low contrast points need to be rejected to improve matching stability.

Applying Taylor expansion on the scale space to determine extreme points and location using Equation (2.12):

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (2.12)$$

The offset from sample point where D and its derivatives are evaluated in Equation (2.13):

$$X = (x, y, \delta)^T \quad (2.13)$$

After derive this function with respect to X and setting it to zero, extremum location is determined in Equation (2.14)

$$\dot{x} = \frac{\delta^2 D^{-1}}{\delta X^2} \frac{\delta D}{\delta X} \quad (2.14)$$

Function value at extremum will be defined as shown in Equation (2.15):

$$D(\dot{\mathrm{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\dot{\mathrm{x}} \quad (2.15)$$

This value is useful for unstable extremum rejection.

The key point descriptor can be represented by allocating consistent orientation to each key point for achieving image rotation invariant. The gradient and direction estimated for an image can be formulated in equation $m(x, y)$ in Equation (2.16) and $\theta(x, y)$ as shown in Equation (2.17) respectively (Mahesh & Subramanyam, 2012):

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + ((L(x, y+1) - L(x, y-1))^2} \quad (2.16)$$

$$\theta(x, y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad (2.17)$$

where L is the scale of key point for image

Orientation histogram is formed by taking consideration of gradient of sample points surrounding key point. Total of 36 bins has covered for 360 degree orientations range. Peak in histogram has almost close similarity to dominant direction of gradients. Other peaks within 80% of highest peak are utilized as key point to that orientation.

2.2.3 SURF Feature Detection

SURF stands for Speeded-Up Robust Features. This algorithm generally can achieve fast computing speed compared with other methods. SIFT is similar to SURF but they are different in terms of key point detection and descriptor extracted. In speed factor, SURF is faster than SIFT (Karahan et al., 2014). Feature point selection and description are the two main parts in SURF.

2.2.3.1 Feature point selection

Theoretically, SURF feature point detection is based on scale space. Hessian matrix based detector is selected. Determinant of this matrix used as discriminant to calculate local maximum value. With σ as scale of image I at point X, Hessian matrix is defined in Equation (2.18)

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \quad (2.18)$$

$L_{xx}(X,\sigma)$ is convolution for mid-point X of image with Gaussian second order filter. Others have similar implication with $L_{xx}(X,\sigma)$. Threshold is set to the extreme points detected. As extreme point larger than threshold value, this will be employed with non-maxima suppression in 3x3x3 three dimensional neighborhoods. Total of 26 adjacent pixels which include 8 adjacent pixels of the same level, 9 adjacent pixels of upper and lower level will be compared. Response of the points which larger than 26 adjacent pixels will be selected as feature point and interpolate in scale space with stable position and scale values. Points detected lesser than threshold are excluded.

2.2.3.2 Feature point description

For extraction of feature descriptor, a square window region with feature point as center is constructed and window is divided into 4x4 sub window regions. In each sub window, Haar wavelet response is computed at 5x5 sampling points. To simplify Haar wavelet response, dx represents response in horizontal direction whereas d_y represents response in vertical direction (Zhu et al., 2014). By summing response of each sub region and response absolute value, four dimensions are formed with $\sum d_x$, $\sum d_y$, $\sum |d_x|$, $\sum |d_y|$ are generated in descriptor. Each sub window has this four dimensional vector, thus a 64dimensional

vector is obtained. Construction process of descriptor is shown in Figure 2.4 (Zhu et al., 2014):

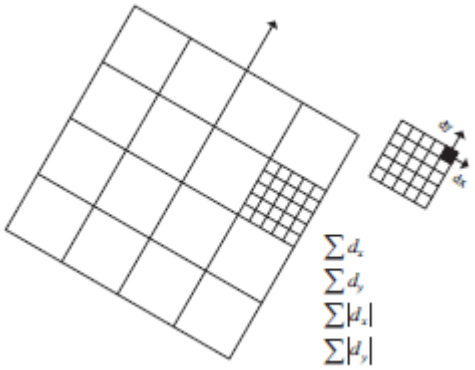

Figure 2.4: Construction for SURF feature descriptor extraction (Zhu et al., 2014).

2.3 Homography Estimation

Homography can be known as projective in which it contains similarity of projective space. In image mosaic, it is used as an estimation to stitch multiple images which have similar key points into single composite image. Estimated homography represents confidence level of correspondence between images (Seo et al., 2013).

Obtaining correspondence is an important part in image stitching as it is used in homography estimation. Generally, there are three main phases for searching the correspondences to estimate homography which listed as below:

i. Extract robust feature points and perform feature matching
ii. Eliminate outlier and obtain best consensus.
iii. Generate homography

After extracted feature point, correspondence between feature points of reference image will be evaluated with feature point of input image. The suitable matching candidate is determined by identifying nearest neighbor in feature vectors data sets from input image. Distance between two neighbor images is calculated and minimized to adjust matrix between neighbor images. Matching candidate must contain feature points with minimum Euclidean distance for invariant descriptor vector. Due to its high dimension and complexity as comparing each single vector, a k-dimensional tree can be used to search all matching pairs.

Random Sample Consensus, RANSAC is a method to find best consensus about correspondence point. With a set of data contains outliers, it is used to estimate parameters of mathematical model (Adel et al., 2014). A homography can be estimated by utilizing Direct Linear Transform (DLT) algorithm or Levenberg-Marquardt (LM) algorithm.

Homography is also a 3x3 matrix to relate reference points with target images and is formulated as Equation (2.19):

$$X = HX' \tag{2.19}$$

where X is reference points and X' is corresponding target points. Matrix H is illustrated in Equation (2.20):

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{2.20}$$

2.3.1 Direct Linear Transform

DLT algorithm can be written in matrix equation as shown in Equation (2.21):

$$\begin{bmatrix} 0^T & w_i' X_i^T & y_i' X_i^T \\ w_i' X_i' & 0^T & x_i' X_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (2.21)$$

Where h^j represents j-th column of matrix H, X_i^T is point(x_i, y_i, w_i) of reference image and X_i' is point(x_i', y_i', w_i') which representing target corresponding point.

To simplify equation, this can be expressed in Equation (2.22):

$$A_i h = 0 \quad (2.22)$$

where A_i is 2x9 matrix with i-th point of correspondences.

2.3.2 Levenberg-Marquardt

LM method is one of the minimization methods. Nonlinear square evaluation is used for bundle adjustment optimization. Important task is to minimize transfer error by using Equation (2.23) (Luo & Gwun, 2010):

$$d = \sum_i (d(X_i, H^{-1} X_i')^2 + d(X_i', H X_i)^2) \quad (2.23)$$

Where d is Euclidean distance, $X_i \; and \; X_i'$ are correspondence points, H is estimated homography for which to be minimized and H^{-1} is inverse of H.

2.4 Affine Transformation

General form of two dimensional affine transformations is defined in Equation (2.24) (Hui & Du, 2010).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s. \begin{bmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.24)$$

Where (x', y') and (x, y) pixel coordinates of the two images respectively.

There are four unknown parameters $(s, a, \Delta x, \Delta y)$ to be solved (Hui & Du, 2010). The expansion for Equation (2.24) is expressed in Equation (2.25) to find unknown parameters in Equation (2.24).

$$\begin{cases} x_1' = s.(\cos(a).x_1 - \sin(a).y_1) + \Delta x \\ y_1' = s.(\sin(a).x_1 + \cos(a).y_1) + \Delta y \\ x_2' = s.(\cos(a).x_2 - \sin(a).y_2) + \Delta x \\ y_2' = s.(\sin(a).x_2 + \cos(a).y_2) + \Delta y \end{cases} \quad (2.25)$$

To accomplish image registration, affine transformation parameters are calculated using coordinates of control points and geometric transformation is performed on images. As transformed coordinate is not an integer in geometric transform, mapping points between target and original image can be acquired by interpolation which include nearest neighbor interpolation, bilinear interpolation and bi-cubic interpolation.

2.5 Image Matching and Blending

An open source library in openCV which known as fast library for approximate nearest neighbors (FLANN) is released and used for nearest neighbor matching. FLANN is a new and fast approximate matching of binary features algorithm (Muja & Lowe, 2014). FLANN Based Matcher provide interface to perform matching using FLANN. The

algorithm works best in fast approximate search in randomized k-d trees which is in high dimensional spaces.

Image blending is one of interpolation type. It is basically to produce an image result with no transition can be identified between original images (Luo & Gwun, 2010). Average weighted blending commonly applied in average image at overlapping area. Bilinear blending can result in ghosting effect, blurring and visible seams. As for linear blending, the result is fast as it is compromise between quality and speed. If unlike ghosting and blurring effect, pyramid blending can be considered and result is better than the linear blending. Another type is multi band blending which able to ensure clean transitions among images without any illumination gaps yet still preserve high frequency details (Luo & Gwun, 2010).

2.6 OpenCV library

Open Source Computer Vision is known as OpenCV and it is an open source programming function library. Generally, OpenCV is an image processing library. It is free for either academic or commercial usage under BSD license (Culjak et al., 2012). There are many libraries developed for C, C++, Python and Java which supports in Windows, Linux, and Mac OS. Originated by Intel, these sources are applied for image and video processing area.

OpenCV library is grouped into modules. Each module contains own structure and feature which devoted to particular computer vision issues. The classes, methods and functions are compiled in cv namespace. Namespace cv is required to declared in order to access required method or function. The main focus is the basic image container known as Mat

class. Generally Mat is a class with two parts. The two parts are matrix header and pointers that contains pixels values. Matrix headers contain attributes about images whereas pixel values depend on methods used for storage (Culjak et al., 2012).

In object class of Mat, there is automated memory allocation. In this instance, the allocation tasks performed when image is read or released. Another unique of Mat class is the reference counting and shallow copy implementation. This is to handle problems the heavy tasks computed by image processing algorithm that contains large number of varied image processing function.

In reference counting system, Mat object has own header. When image is assigned to the other image, data is not copied but only headers are copied by operators. Both images point to same address or memory block.

Figure 2.5 shows the sample code with explanation.

```
Mat X, Z;                                   // Header created
X = imread(argv[1], CV_LOAD_IMAGE_COLOR);   //Allocation matrix used
Mat Y(X);                                   //Copy constructor
Z=X;                                        //Operator assignment
```

Figure 2. 5 Sample Code

Pixels are the basic image contents or picture elements in digital image. They are classified as smallest addressable point in images. An efficient processing pixel is important since

images might made of thousands pixels. In OpenCV, element of matrix can be represented as single pixel in matrix which is the basic data structure.

Images are grouped into two levels. First one is the gray level image which typically in 8bit pixel values where as another is the color image (Red, Green and Blue) in 8bit unsigned values per pixel to illustrate their colors respectively. Generally, various value types are allowed in OpenCV but certain pixel processing only can be applied for particular matrix types. Color space selected can affect certain image processing. Row and column are specified in order to access individual matrix element and few methods can be called for accessing (Culjak et al., 2012).

One of the method is *at(int x, int y)*. The return type must be identified at compilation time. For example, coordinates of pixel on *(j,i)* of Mat object image with return type as 8 bit unsigned value, can be coded as *image.at<uchar>(j,i)*. Another method is accessing pixel elements with pointer which is much more efficient. Direct accessing to address of image row can be performed using ptr method, eg. *uchar* data = image.ptr<uchar>(j)*. For arithmetic image combination, numerous functions such as add, absdiff, multiply, divide, etc. can be used to perform operation and purposes requested.

2.7 Current work on image stitching

Image stitching is a common active research and challenging field in computer vision (Adel et al., 2014). This involves image processing and several processes to obtain a smooth and seamless output result. Several researchers study and describe feature based

technique to make improvement on detecting feature which is part of the process to produce panoramic image.

Patel, (2012) provides some basic introduction on techniques and algorithms that able to generate panoramic image. Some of the corner detection algorithms include Harris Corner, Smallest Univalue Segment Assimilating Nucleus (SUSAN) (Smith & Brady, 1997), Forster and SIFT algorithms are described in the paper. Fundamental of image mosaicing is also presented in the paper.

Jain et al., (2012) shows an improved Harris corner detection method by extracting corner without manually setting threshold. Corner extracted is sensitive to noise, edges and point isolation. This method is included in image mosaicing algorithm to provide panoramic image. In his technique, two corners of two images are detected firstly. Next false corners are removed from images and homography is estimated to find matched corners.

SUSAN corner detection is used in (Yanli et al., 2008) to extract feature point. Translation parameters of two images are estimated using phase correlation techniques. Then RANSAC algorithm is included to delete wrong matching points after initial matching. Based on epipolar and homography constraints, fundamental and homography matrixes are estimated. For image fusion, linear weighted transition method is used. Output results show robustness and precise.

Zetao et al., (2012) used SIFT feature detection and algorithm to correct image distortion and de-ghosting image. Distortion occurs during obtaining images using standard camera or due to camera optical axis. As distortion correction executed, feature points are matched and image stitching is performed. Proposed SIFT with openCV able to reduce ghosting to certain extent.