Mosaicking Video with Parallax

CHEUNG Man-Tai

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

In

Mechanical and Automation Engineering

August 2001

Abstract

Image Mosaic construction is about stitching together adjacent images of a scene into an image that displays a wider field of view. Existing mosaic construction algorithms have the following restrictions: (1) the scene is planar or very distant, or (2) the scene is generic but the camera motion is a pure rotation. In either case the registration of images involves a global parametric mapping. In this work we address the general case: a generic scene is pictured under an arbitrary camera motion. The image data so captured contains parallax that makes the registration of images a difficult problem because global parametric transformation that allows one image to be registered with another is lacking. The work is based upon a 3-image algorithm that is capable of constructing mosaic from images of such nature. The algorithm exploits projective reconstruction to solve for the problem of geometrical image transfer between the images being registered. We extend the algorithm for an image stream that contains parallax. We answer two questions: (1) how an image stream is divided into various 3-image sets for the three-image algorithm to iterate upon; and (2) how intermediate mosaic results over the various 3-image sets are accumulated to compose the overall mosaic at the end. The framework allows uneven sampling of the video stream. Experimental results on real image data are presented.

摘要

相片鑲嵌製作是將同一場景的不同照片組合成一張有較大視野的相片，現存的相片鑲嵌技術有著以下的限制：(1) 場景須要是平面或遠離鏡頭，或 (2) 一般場景但鏡頭動向只能作純旋轉， 以上情況會利用一個全體性參數拼合相片。在這裡我們嘗試一般情況：利用一個隨意活動的鏡頭拍攝一般場景，在得出的相片中會發現視差，視差會令拼合相片成爲一個難題因缺乏一個全體性參數以作拼合相片。這裡的研究工作是建基於一個名爲三相片系統的技術，這個三相片系統有能力應付上述情況。這個新技術是利用投射性重組，以解決拼合相片間的幾何移送問題。我們更將這個技術延伸至包含視差的相片串。在這裡我們解決了兩個問題 (1) 如何將相片串分折成一組三相片以應用*三相片系統*， 及 (2) 如何將過程中得到*三相片系統*鑲嵌相片累積成最後的鑲嵌相片。這個技術也容許對錄像的不均等取樣. 這裡也提供對真實照片的實驗成果。

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Background

Image mosaic construction is about stitching together adjacent images of a scene into an image that displays a wider field of view. Image mosaicking can be used in a wide variety of applications including remote sensing, visual surveillance, virtual reality, and video compression.

There have been a few pieces of work about it in recent years [8,11,14,16], but they are only effective in constrained situations where the camera motion is a pure rotation or the viewed scene is planar or very distant. The more general case of a generic scene viewed under a general camera motion has not been addressed much. What makes the general case more difficult is the parallax present in the images. Mosaicking images with parallax is challenging because there is no global parametric transformation between the images, as is present in the above restricted cases [8,11,14,16], that allows the images to be registered and warped to the same image frame.

1.1.1. Parallax

Parallax [18] is the distortion in the image that is resulted from the displacement of the camera motion. As shown in Fig. 1, if two distinct 3D points (*P* and *Q*) project to the same point in an image Π *'* (whose optical center is *O'*) along the same view ray. There would exist a displacement in their image positions if the viewpoint changes. For example if we change the viewpoint from Π *'* to another image Π (whose optical center is *O*) the image *p'* on Π *'* will become two image points *p* and *q* on Π . That displacement between the two image points is the parallax that we are addressing.

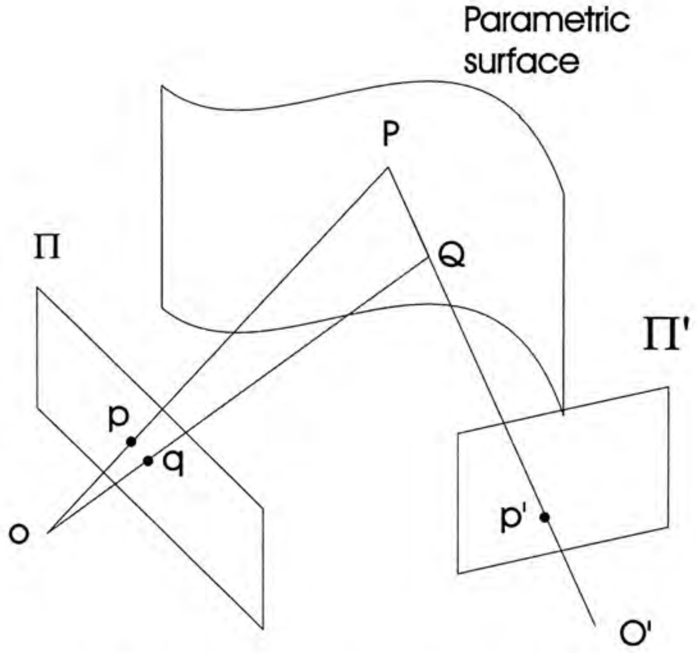


Fig. 1 : Illustration of parallax.

The existence of parallax in images makes image mosaicking problematic because of the lack of global parametric transformations between the images. Blurring and ghosting will occur if we use a global transformation to construct mosaic from images that contain parallax. For example, consider the case in Fig. 1. If we transform the image Π ' to image Π using a global transformation, blurring and ghosting will exist since p' on image Π ' will transform to p and q on image Π.

1.2. Literature Review

There have been a few pieces of work about mosaic construction in recent years [8,11,19,20,26]. Global parametric transformations between two images are typically used to register the images and thereby construct the mosaic. The transformations include image-plane similarity transformations, affine transformations, bilinear transformations, and planar-projective (homography) transformation.

The work [26] demonstrated the use of homography as the transformation to register images. It can construct 2D mosaics from image with small overlap and is able to deal with cases when the rotation around the optical axis and zooming are large. The work [12] uses Gaussian Pyramid to construct mosaic. It has the ability to do inverse mosaic for tracking the current image frame, but the work is limited to panoramic scene. The work [16], unlike most of the other works that mainly focus on restrictive situations. It allows zooming and forward motion in the camera, but it relies on the use of some nonlinear method. The work [23] focuses on constructing panorama from image sequence. It uses planar-projective transformation to register adjacent image frames instead of projecting the images onto a common surface. It then refines the mosaic result using global and local alignments.

The work [7] aims to construct real-time video mosaicking from images of the ocean floor captured by a semi-autonomous underwater vehicle. The work uses kalman filter and correlation method between image frames to reduce the error in resulting mosaic. It shows the ability of reducing the propagation of image alignment errors within the mosaic and the use of resulting mosaic for the position estimate of an underwater vehicle. But the effect of parallax to mosaic construction was not mentioned in this work.

In [19] authors proposed a method in which both the planar and parallax motion components are computed with a coarse-to-fine scheme, but a highly complicated nonlinear system is required to deal with a large number of unknowns in order to obtain a good result.

In summary, most of the previous works are effective only in constrained situations where the camera motion is a pure rotation or the viewed scene is planar or very distant. The general case, i.e., the case of having a generic scene viewed under a general camera motion, has not been addressed much. What makes the general case difficult is the parallax present in the images. Mosaicking images with parallax is challenging because there is no global parametric transformation between the images, as is present in the above restricted cases [14,16,19,20,26], that allows the images to be registered and warped to the same image frame for the stitching process.

A framework proposed in [3] allows mosaic to be constructed from images with parallax. The work argued that in such a case two images could not possibly allow mosaic to be constructed. It proposed the use of an additional image, termed the intermediate image, that allows intensity region visible only in one image but not in the other to be warped from the former to the latter. The idea was to perform projective reconstruction from feature matches of the images to obtain 3D projective shape. Then project the 3D projective shape onto the mosaic frame. The merit is that all the techniques used in the system are classical techniques in computer vision. However, the framework applies only to a discrete set of three images. As the framework involves not two adjacent images but three images, it is not obvious how the framework is extended to process not three images but an image stream.

1.3. Research Objective

The first aim of the research is to study the 3-image algorithm first proposed by [3] and making modification on the existing system of the 3-image algorithm. We also aim at developing a mechanism that allows the framework to be applied toward an image stream. The mechanism should allow image mosaicking to be no longer limited to planar scene or pure rotation of camera. Mosaic can be constructed even under arbitrary motion of a hand-held camcorder. Experiments are also conducted in order to support the arguments in developing the system.

1.4. Organization of Thesis

The remainder of the thesis is organized as follows. Section 2 reviews the 3-image algorithm. Section 3 discusses the extension of the algorithm from the case of a discrete image set (the 3-image algorithm) to the case of an image stream (the n-image algorithm). Section 4 shows the modifications on the n-image algorithm to allow the system to produce a more pleasing mosaic using an uneven-sampling-rate of the input video. Section 5 presents experimental results we obtained in applying the proposed method to real and synthetic image data.

Chapter 2. The 3-Image Algorithm

Given two images (that are taken from different viewpoints) of a generic scene, we aim to construct a mosaic that displays all that is visible in the two images. In this work, all image points will be represented by their homogeneous coordinates. Let R and t be the rotation and translation components of the spatial transformation between the two camera coordinate frames. Let K denote the intrinsic parameters of the camera (a 3×3 upper-triangular matrix). For any pair of matched pixels or features, p_1 and p_2, in the two images, we have the following equations [5,6,9,12,18,21]($\cong$stands for the equality up to a scale factor):

$$p_2 \cong \underbrace{KRK^{-1}p_1}_{\text{planar}} + \underbrace{\frac{1}{z}Kt}_{\text{parallax}} \tag{1}$$

where z is the depth of the corresponding 3D feature. Since KRK^{1} is the homography at infinity, and Kt represents the epipole e_2 on the second image, Eq.(1) can be written as:

$$p_2 \cong H_{\infty} p_1 + \frac{1}{z} e_2$$

Thus the 2D motion of the feature or pixel can be decomposed into two components (Eq.(1)): (i) planar (the first term in the above equation), and (ii) parallax (the second term in the above equation). Note that this decomposition can be done with respect to any arbitary plane Π (real or virtual) in the environment [11]. The parallax is the image projection of the deviation of the associated 3D feature from the chosen plane.

The above equation can be written as

$$p_2 \cong H_{\Pi} p_1 + k e_2 \qquad (2)$$

where k can be considered as the projective depth of the point p_1. In this case, the parallax is defined with respect to plane Π.

While the planar transformation can be computed by choosing a physical or virtual plane in the scene, the second component depends on both the camera translation and the individual depth of the considered pixel.

From Eq. (2), one would notice that the knowledge of the correspondence (p_1, p_2) and the knowledge of the scalar k are equivalent in the sense that the knowledge of one yields that of the other. However, due to the intrinsic property of images that their texture might not permit a pixel-to-pixel correspondence for the whole image, the parallax component is not known for the majority of pixels even though it is known for some of them (the feature correspondences). As a consequence, one cannot use Eq. (2) to register the two images.

It is impossible to construct a global 2D parametric transformation between two images in the general case (an arbitrary scene under an arbitrary camera motion), so we make use of a third image which we refer to as the intermediate image.

We are thus left with three images. We call them as follows. The *reference image* R is the image whose viewpoint is where all image data are warped to, and where the final mosaic is constructed. The *target image* T is the image to be warped to the viewpoint of the reference image for constructing a mosaic there. The *intermediate image* I is a third image that is to assist the warping of the target image to the reference image; it should show something in common with the target image as well as with the reference image. It should be noticed that we have two parallax fields: (i) the one associated with the couple target-reference, and (ii) the one associated with the couple target-intermediate.

2.1. Projective Reconstruction

Given a pair of uncalibrated images with no knowledge of the intrinsic and extrinsic parameters of the camera, a set of pixel correspondence is the only information that we have and we do not know the location of the original 3D points. What we can recover from this set of correspondence is the camera transformation matrices (i.e. the fundamental matrix) and the point placements may be determined up to a collineation of projective 3-space (only the connection of points can be recovered but the size, angle and parallelism cannot be recovered) [4,8].

Thus, our 3 images are related to an arbitrary projective space. Each image will have a 3×4 projective mapping M that maps 3D projective space into the image plane such that

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where *[X Y Z 1]'* are the 3D homogenous coordinates of a object point and *[u v 1]'* are the homogenous coordinates of an image point on the image plane. Let *M, M'*,and *M"* be the projective mapping associated with the target image, the intermediate image, and the reference image respectively. These three matrices can be easily inferred from point matches. But before recovering the projective mappings, we have to recover the fundamental matrix between the target image and the intermediate image.

2.2. Epipolar Geometry and Fundamental Matrix [5]

Any two images of a single scene/object are related by the epipolar geometry. It is independent of scene structure, only depends on the cameras' intrinsic parameters and relative pose. Consider the case of two cameras as shown in Fig. 2. Let *O* and *O'* be the optical camera centers of the first and second cameras. Given an object point *P* projected onto the image planes *i* and *i'*, we will get the image points *p* and *p'* lying on the two image planes *i* and *i'* respectively. For image point *p* on the image plane *i*, its corresponding point in the second image is constrained to lie on a line *l'* called the *epipolar line*. The epipolar line is the intersection of the plane Π defined by *P*, *O* and *O'* (known as the *epipolar plane*) with the second image plane. The image point *p* may correspond to an arbitrary point on the line *OP* (*P* may be at infinity) and the projection of *OP* on the second image plane is the epipolar line. Moreover, the intersection of the line *OO'* and the image planes form the epipoles (*e* and *e'* respectively) of the images. If *p* and *p'* correspond to a single physical point *P* in 3D space, then *p*, *p'*, *O* and *O'* must be coplanar.

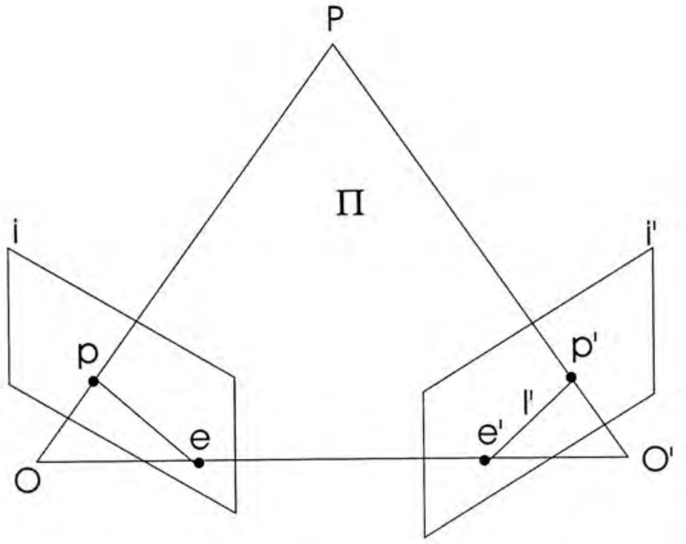


Fig. 2 : The epipolar geometry.

Epipolar geometry can be described by a 3×3 singular matrix called the fundamental matrix. It captures all geometric information contained in the two images such that if we have 2D point *x* in the first image and *x'* in the second, then the image points satisfy the relation $x'^T \; Fx = 0$.

2.3. Determine the Projective Mappings

Let *F* be the fundamental matrix between the target image and the intermediate image, and *e'* be the corresponding epipole in the intermediate image. It is well known that a solution for the mappings *M* and *M'* is given by [22]:

$$M \cong [I \quad 0]$$

$$M' \cong [S(e')F + e'w^T \quad \omega e']$$

for some 3-vector *w*, and a non-zero scale ω. Matrix *I* represents the 3×3 identity matrix, *S(e')* is the skew-symmetric matrix associated with the 3-vector *e'*.

Once *M* and *M'* are determined, the 3D projective coordinates of all feature matches present in the target image and the intermediate image can be recovered. The third mapping *M"* is then obtained by imposing that some reconstructed 3D points are reprojected to their matches in the reference image frame. The criterions are presented in the coming section.

2.3.1 Conditions for Initial Matches

The fundamental matrix *F* between the target image and the intermediate image is a 3×3 matrix of rank two. Also, it is defined up to a scalar factor. Therefore, a fundamental matrix has only 7 degrees of freedom. There are only seven independent parameters among the 9 elements of the fundamental matrix. As each point correspondence provides a linear equation for the entries of F, so it is possible to recover the fundamental matrix using 7 pairs of correspondence points between the target image and the intermediate image. In our system, we use the linear least-squares technique to recover the fundamental matrix. In other words, if we ignore the dependence of the 8 degrees of freedom, we can recover F (up to an overall scale factor) by a linear estimation more easily with at least 8 point correspondences.

Suppose we have *n* point correspondences, where $n \geq 8$, we will get the following homogeneous system of linear equations

$$B\vec{f} = \vec{0}$$

$$\underbrace{\begin{bmatrix} x_1x_1' & y_1x_1' & x_1' & x_1y_1' & y_1y_1' & y_1' & x_1 & y_1 & 1 \\ & & & & \vdots & & & & \\ x_nx_n' & y_nx_n' & x_n' & x_ny_n' & y_ny_n' & y_n' & x_n & y_n & 1 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{32} \\ f_{33} \end{bmatrix}}_{\vec{f}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\vec{0}} \tag{3}$$

where B is a $n \times 9$ matrix related to the image coordinates of feature correspondences in the target image (*[x y 1]*^{T}) and intermediate image (*[x' y' 1]*^{T}). $\vec{f}$ is a vector associated with the elements of the fundamental matrix. In order to recover a nontrivial fundamental matrix, the rank of *B* must be equal to 8. If the rank of *B* is less than 8, there will be too many degrees of freedom in the fundamental matrix other than the arbitrary scale factor. If the rank of *B* is 9 (an $n \times 9$ matrix cannot have rank more than 9), the system of equations is just determined and the only solution we get from the system of equations is the trivial solution which does not conform to physical situation [8,10,14].

The projective mapping *M"* of the reference image is a 3×4 matrix which is defined up to a scalar factor. Therefore, a projective mapping has 11 degrees of freedom. There are only 11 independent parameters among the 12 elements in the projective mapping. Since each point correspondence will generate two equations in recovering the projective mapping, there must be at least six matches between the feature points on the reference image and the 3D points reconstructed from target image and intermediate image, so that we have at least 12 equations to estimate 11 degrees of freedom of the projective mapping *M”*

To recover the projective mapping *M”* using the least-squares technique, we have the following homogeneous system of equations.

$$A\vec{x} = \vec{0}$$

$$\underbrace{\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_nX_n & -x_nY_n & -x_nZ_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_n & -y_nZ_n & -y_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{33} \\ m_{34} \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\vec{0}} \quad (4)$$

where *A* is a $2n \times 12$ matrix associated with the coordinates of the reconstructed 3D points (*[X Y Z 1]*$^{\mathrm{T}}$) and image points in reference image (*[x y 1]*^{T}), and *n* is the number of point correspondences used to recover the projective mapping. $\vec{x}$ is a vector associated with the elements of the projective mapping *M”*.

In order to get a non-trivial solution of *M”*, the rank of *A* must be equal to 11. If the rank of *A* is less than 11, there will be degrees of freedom beyond that of the arbitrary scale factor of *M”* cannot be recovered. If the rank of *A* is 12, the system of equations is just determined, and we can only get a trivial solution for *M”* but that is physically impossible [8,10].

In summary, the initial matches must fulfill the following conditions:

(1) On the feature correspondences between the target image and the intermediate image:

a) The number of initial matches must be at least 8 (assuming the linear least-square method is used).

b) The matrix B formed by the initial matches (according to Eq. (3)) must be of rank 8.

(2) On the feature correspondences between the target, intermediate, and reference image:

a) The number of initial matches must be at least 6.

b) The matrix A formed by the initial matches (according to Eq. (4)) must be of rank 11.

To get such matches we use the software developed by Zhang et al. [25]. Details of the software will be discussed in the following section.

Our method relies on the following fact. In general, for any pixel of the image to be registered (target image) if we know the 2D location of its correspondence in the intermediate image we are able to transfer this pixel to the reference image using a projective reconstruction followed by a projection, i.e., using the three projective mappings M, M', and M''. This is illustrated in Fig. 3.

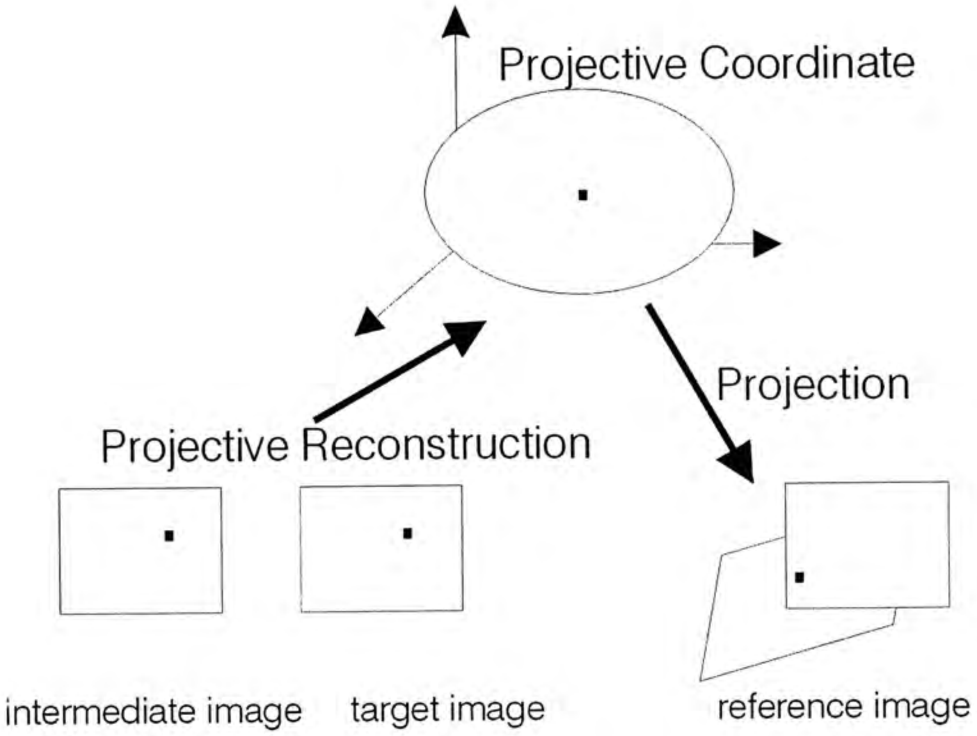


Fig. 3: Illustration of the three-image algorithm.

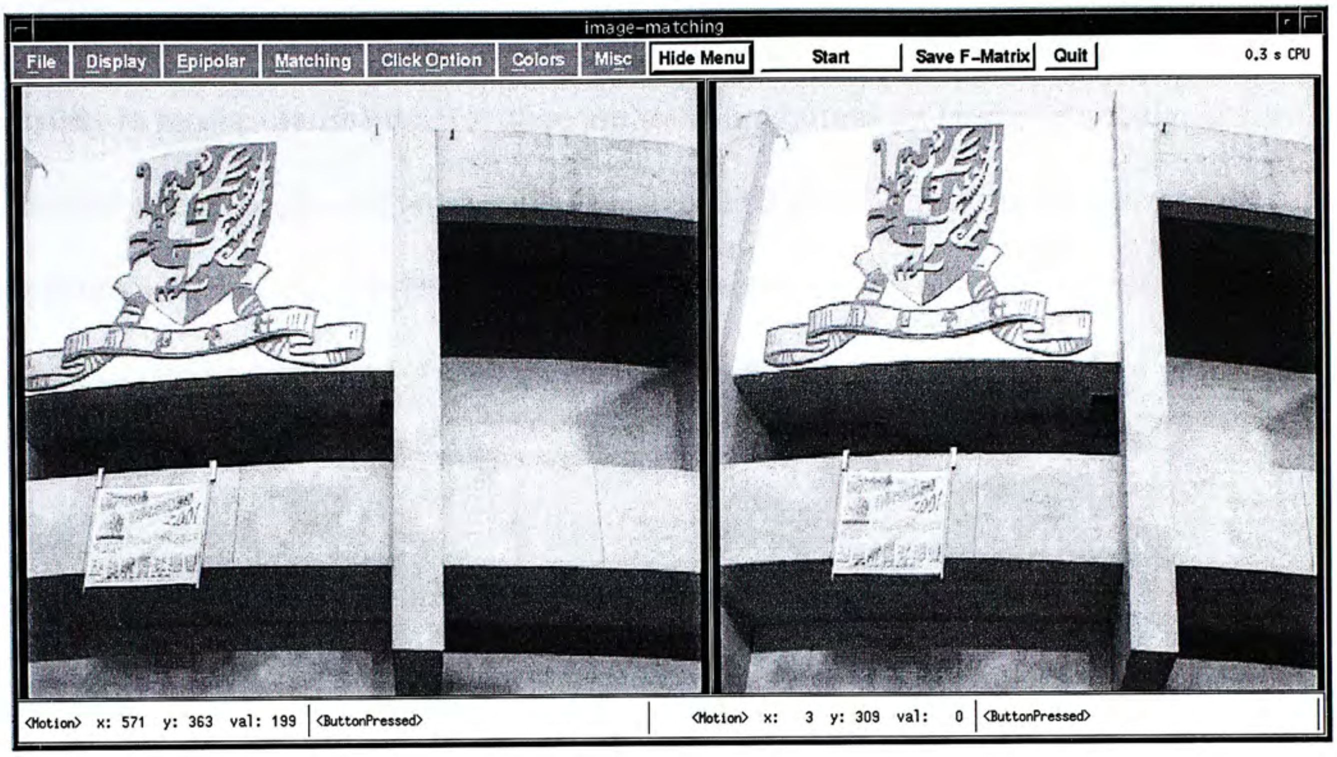
2.3.2 Obtaining the Feature Correspondence

We use a software developed by others but not our own because the main purpose of this research work is develop an algorithm for image mosaicking but not feature matching. So the focus was on the mosaicking part. Moreover, it is very difficult to include the feature matching in the research work because feature matching itself is a challenging work. It caused many researchers lots of time in developing system for feature matching. If we construct our own feature matching system, its preference might not as good as the system developed by the others. So it would be more efficiency to use a feature matching system that developed by other researchers.

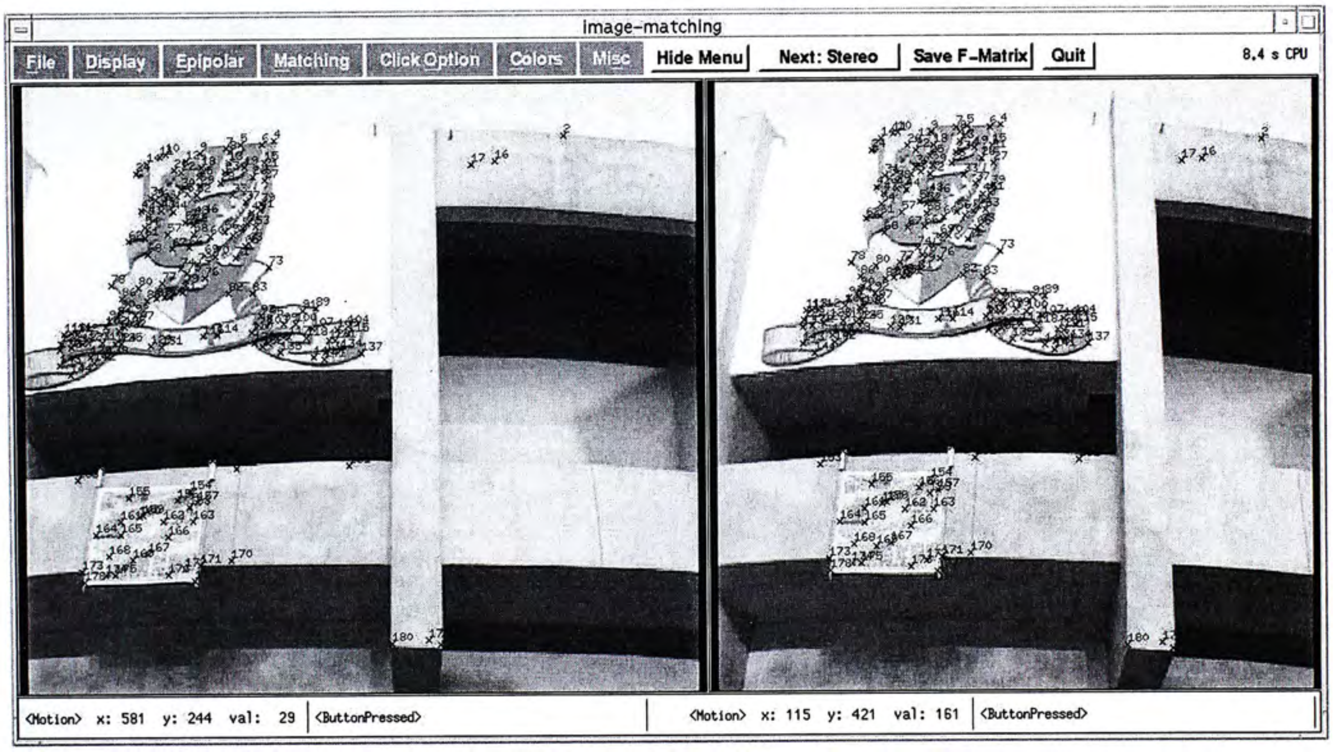
Since the quality of the feature correspondence plays an important role in our research work, there are some requirements about the feature matching system. The accuracy of the correspondence feature is very important since it will affect the error of the estimation of the transformation mappings used in the mosaicking algorithm. The feature matching system should also able to work with uncalibrated images because we do not want to add limitation on the camera used in capturing image/video.

The software *image-matching* developed by Zhang et al. [23] was used to obtain the matches that to be use in the 3-image algorithm. *Image-matching* is a software which implements a robust technique for binocular image matching by exploiting the only available geometric constraint, the epipolar constraint. It thus computes also the epipolar geometry, in terms of the fundamental matrix, between two images.

If the images are uncalibrated, the motion between them and the camera parameters are not known. Thus, the images can be taken by different cameras or by a single camera at different time instants. If we make an exhaustive search for the epipolar geometry, the complexity is prohibitively high. The idea underlying the approach of *image-matching* is to use classical techniques (correlation and relaxation methods) to find an initial set of matches, and then use a robust technique---the Least Median of Squares (LMedS)---to discard false matches in this set. The epipolar geometry can then be accurately estimated using a meaningful image criterion. More matches are eventually found, as in stereo matching, by using the recovered epipolar geometry. Fig. 4. shows an example of using *image-matching* to obtain feature correspondence.



(a) Input image pair



(b) corresponding features obtained

Fig. 4.: example of using *image-matching* to obtain feature correspondences from a pair of images

Although *image-matching* is able to match corresponding features with acceptable quality in most experiments. But there are some limitations on *image-matching*. If the features of the scene were not distinguished enough, it will affect the accuracy of the resulting matches. Moreover, *image-matching* uses points obtained by corner detector to perform feature matching. If such corners are not available in the images, it also affects the accuracy of the resulting matches.

2.4. Registering Pixel Element

Since parallax also exists between the target image and the intermediate image, so transferring pixels from the target image to the intermediate image for the entire image is as difficult as transferring them onto the reference image. Recall the situation that when parallax is present in an image pair, the correspondence pixel can be represented by:

$$p_i' \cong Hp_i + d_i e'$$

where p_i' denotes pixel on the intermediate image and p_i represent pixel on the target image. H is the planar motion associated with the reference plane (the plane is chosen as the average plane associated with all feature correspondence in the target image and the intermediate image), and e' represent the epipole of the intermediate image. Both H and e' can be recover from the feature correspondences between the target image and intermediate image.

Since the parallax d_i is unknown for each pixel, the above equation cannot be used to transfer the target image pixels to the intermediate image. Therefore, it is necessary to set the parallax d_i to an approximated value $\overline{d}_i$, since the exact value of parallax of every pixel cannot be recovered for the reasons mentioned above. Transferring target pixels to the intermediate image is performed using the following equation:

$$p_i' \cong Hp_i + \overline{d_i} e'$$

Approaches that proposed to determine the approximation will be presented in the following section

2.4.1. Single Homography Approach

First taking $\bar{d}_i = 0$. Then the above equation will become

$p_i' \cong Hp_i$

this is a good approximation if the distance between the centers of projection associated with the target image and the intermediate image is small compared to the scene depth.

One might question that this approximation would mean the camera motion between the target and intermediate images is a pure rotation, and a pure camera rotation would not allow any 3D notion about the scene to be recovered. However, the approximation, as a planar homography, also covers the case that there is considerable translation between the images but the scene is planar. So we can use the latter interpretation in introducing the mapping, approximating the scene with a plane (the reference plane) that best fit the available feature correspondences. So the approximation of the transformation between the target image and the intermediate image can be obtained.

Once p_i' is obtained, the 3D projective coordinates of the 3D point that projects onto pixel p_i can be easily computed from the two projective mappings M and M'. Then, the 2D location in the reference image (mosaic frame) is computed by the 3D point using the projective mapping M''.

The Single Homography approach is able to estimate p_i' when the scene can be approximated by a reference plane. But if the scene cannot be approximated by a reference plane. A great error will be the result of using Single Homography approach. Another approach was introduced to overcome the problem.

2.4.2. Multiple Homography Approach

In this approach we taking $\overline{d}_i$ = closest parallax which is known. Recall the approximation of the first case, that the scene can be represent by a reference plane. Now we do not assume the scene can be represented by one single reference plane but by multiple reference planes. So the approximation will become

$$p_i' \cong H_{local} p_i$$

where H_{local} is a local planar homography that can be calculated from at least four feature correspondence pairs.

For every pixel on the target image, we will search for the three closest features of the pixel. With the help of epipoles of the image, we will get four pair of feature correspondences. We can use the four point pairs to calculate a homography, since four pairs are the minimum requirement for calculating a planar homography. Then using that homography we can transform a pixel from the target image to intermediate image.

We then will perform projective reconstruction on p_i and p_i' to determine the 3D projective coordinate then project it to the mosaic frame with M''.

But this approach comes with some disadvantages. First we need to perform the search of closest features and calculate the local homography once for every pixel. It is a time and resource consuming process. One more important disadvantage is that we cannot ensure that the three closest feature correspondences will give us the correct approximation. In a case if the three feature points are collinear or close to collinear the recovered reference plane will deform to a line in the 3D space. The recovered homography will not give us the correct transformations. Because of all these problems, we had introduced another approach to replace the Multiple Homography approach.

2.4.3. Triangular Patches Clustering

We all know that if you have three distinct points in the 3D space, you can define a plane with the three points. Therefore we can approximate the scene using planes defined by any three 3D points that obtained by the projective reconstruction. But the problem is that if we pick up three points randomly, using them to define a (local) plane in the scene. The approximated scene would not be accurate. So we need an algorithm to cluster the points to found an optimal solution. Delaunay triangulation is one of the ways to do this. Delaunay triangulation is widely used in representing 3D data in the field of computation geometry. It is one of the fastest triangulation methods with relatively easier implementation, giving excellent results for most applications.

2.4.3.1. Delaunay Triangulation

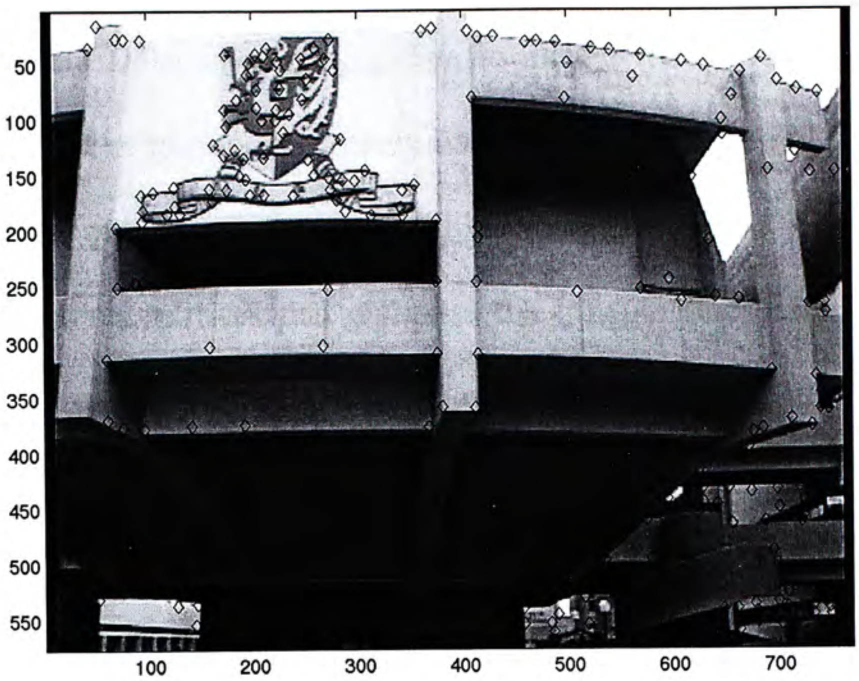
Given a set S of n points s_i of an Euclidean space ε, the correspondence Voronoi diagram is set V of n convex polyhedra V_i where V_i is the set of points which are closer to s_i in ε than to any other point in S:

$$V_i = \{x, x \in \varepsilon, \forall j, 1 \leq j \leq n, d(x, s_i) \leq d(x, s_j)\}$$

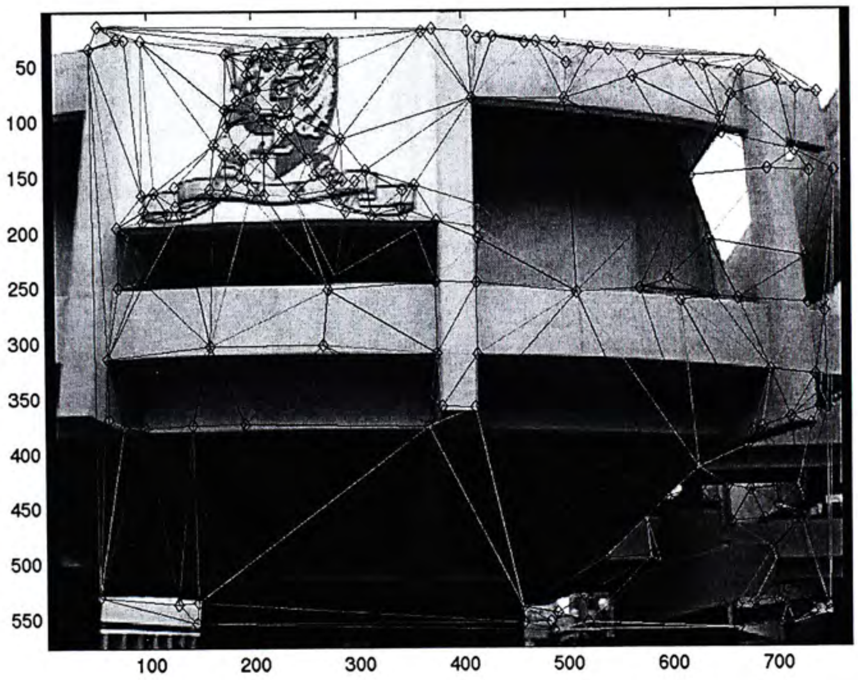
where d denotes the euclidian distance.

The straight-line dual of the Voronoi diagram, obtained by linking line segments of the points s_i, s_j, s_k whose Voronoi polyhedra are adjacent, is called the Delaunay triangulation.

The Delaunay triangulation of a point set $S = \{ s_1, s_2, \ldots, s_n \}$ is defined by the empty circle condition: a triangle $s_i s_j s_k$ appears in the Delaunay triangulation if and only if its circumcircle encloses no other points of S [1,17]. Fig. 5. shows a simple demonstration of the Delaunay triangulation, Fig. 5(a) shows an image of a scene with feature points obtained by *image-matching*. We want to cluster the feature points into triangular patches using Delaunay triangulation. Fig. 5(b) shows the resulting triangular patches.



(a) The input points



(c) The triangular patches obtained by Delaunay triangulation

Fig. 5. : A simple demonstration of Delaunay triangulation.

So we will use the Delaunay Triangulation to cluster the feature correspondence in target image into a set of triangular patches. Each triangular patch will define a 3D plane in the projective space. The equation of this plane is inferred from the projective coordinates of the three vertices that compose the triangle (these are computed at the stage of projective reconstruction). Each plane is parameterized by a four-vector r_j such that:

$$r_j^T x = 0$$

where the four-vector x represents the 3D projective coordinates (up to a scale factor) of any point on the plane.

There is a step different in construct mosaic using the single/multiple homography method and the triangular patches clustering method. If we use the single/multiple homography method, For a pixel p in the target image, we will first find its corresponding coordinate p' in the intermediate image then reconstruct the corresponding 3D coordinate P and find out the corresponding coordinate p'' in the reference image.

For the triangular patches clustering method, For a pixel p in the target image, we will recover its 3D projective coordinate P by computing the intersection of the line of sight and the triangular patch that contains p. P can be computed be solving:

$$p \cong MP$$
$$r_j^T P = 0$$

The above equations provide 3 linear equations in $P = (X, Y, Z, 1)^T$. Since $M = (I\ 0)$ solving P is very simple.

The corresponding coordinate p'' in the reference image can be recovered by using the projective mapping M'' in the way:

$$p'' \cong M''P$$

The occlusion and disocclusion problem is also a problem of the developed algorithm. The quality of the resulting mosaic will be affected by the occlusion and disocclusion problem since it will cause misalignment in the mosaic frame. This issue being one of the most difficult issues in computer vision, is not in this work. However, the fact that we represent a generic scene as a number of small planar patches would reduce the likelihood of this error, in comparison with the global homography method.

The triangular patches clustering method might not be as reliable as the single homography method when the scene is a planar or distant. This is because in the method only three points are used to determine a (local) plane in 3D space and it is thus sensitive to noise in the feature correspondences. But for the single homography method, it assumes a single plane for the entire scene, and all feature correspondences are used to determine just a plane in 3D space. The effect of noise in feature correspondence is thus reduced in this case.

2.5. Mosaic Construction

Once we have approaches to deal different cases, we have to decide which approach should be applied. A simple method was used to decide whether the scene is a planar one or not. The method is that we first calculate the homography using the all feature correspondences of the target image and the intermediate image and measure the error of the homography. If the error is smaller than a threshold value, the scene can be approximated by a global reference plane and the single homography approach will be used. But if the error is larger than a threshold value, the scene cannot be approximated by a reference plane and the triangular patches clustering method will be used.

The final mosaic will be constructed by merging two images: the wraped image (previous section) and the reference images. Fig. 6. shows a summery of the 3-image algorithm.

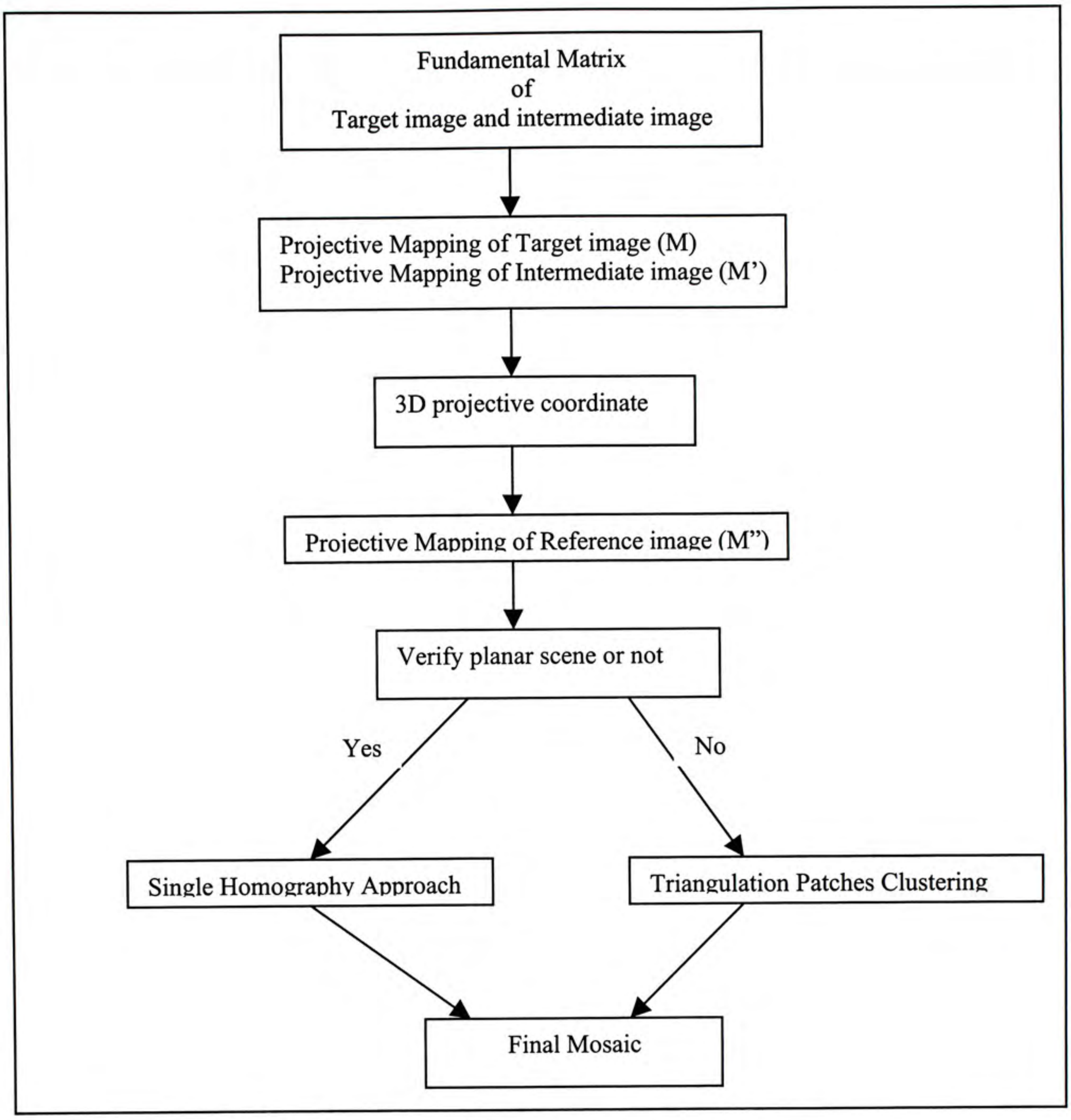

Fundamental Matrix
of
Target image and intermediate image
Projective Mapping of Target image (M)
Projective Mapping of Intermediate image (M')
3D projective coordinate
Projective Mapping of Reference image (M")
Verify planar scene or not
Yes
No
Single Homography Approach
Triangulation Patches Clustering
Final Mosaic

Fig.6 : Summery of the 3-image algorithm.

Chapter 3. The n-Image Algorithm

Here we describe how we extend the 3-imgae algorithm form the case of discrete set to the case of image stream [2].

It could be expected that if the 3-image algorithm is to be extended for an image stream, it would involve (1) iterations of processing over the various 3-image sets of the image stream, and (2) propagation of the intermediate mosaic results across the 3-image sets and at the end to the final mosaic frame. The issues are, how should the image stream be split into various 3-image sets, how the 3 images in each iteration be designated as the image frames *{T,R,I}* in the 3-image algorithm, and most importantly how the mosaic results can be accumulated across the iterations and be propagated to the mosaic frame. In this work we propose a solution to all these questions. The solution contains an orderly splitting of the image stream into 3-image sets as well as a systematic designation of the three images in each set as the *T,R,I* frames. Most importantly, it requires no explicit propagation of intermediate mosaic results across the iterations; all intermediate mosaickings happen at the final mosaic frame.

We first sample the image stream with an equal sampling. We refer to the most current image frame of all these sampled images as *S(t)*, where *t* represent the current time frame, and the second most current image frame as *S(t-1)*, and the third most current image frame as *S(t-2)*, and so on. We assume that the desired mosaic frame is the most current image frame *S(t)*. In other words, we are to warp all the previous images to the most current image frame and construct a mosaic there.

We begin the iterations from the most current end of the image stream. In the first iteration, we pick the images *S(t),S(t-2),S(t-3)* to apply the 3-image algorithm. *S(t)* is designated as the reference image frame *R*, *S(t-2)* as the target image frame *T*, and *S(t-3)* as the intermediate image frame *I*. *S(t-1)* is not used as the target image as very often it resembles the reference image *S(t)* too much and its information content does not justify the mosaicking effort. Using the 3-image algorithm, whatever visible in both *S(t-2)* and *S(t-3)* but not *S(t)* will be warped to *S(t)* to create an intermediate mosaic there. Notice that this mosaic of iteration 1 is constructed at the final mosaic frame *S(t)*. Notice also that in this iteration we have compute a mapping that allows any feature point in *S(t-2)* to be mapped to *S(t)*, the final mosaic frame.

In the second iteration, we pick the images *S(t-2),S(t-4),S(t-5)* to apply the 3-image algorithm, this time with *S(t-2)* as the reference image frame *R*, *S(t-4)* as the target image frame *T*, and *S(t-5)* as the intermediate image frame *I*. However, instead of constructing the intermediate mosaic for these three images at the frame *S(t-2)*, we first make use of the mapping from *S(t-2)* to *S(t)* we have calculated in the previous iteration, to transfer the initial set of feature points over the frames *S(t-2),S(t-4),S(t-5)* to a set over the frames *S(t),S(t-4),S(t-5)*. With this transfer, we have initial matches over not *S(t-2),S(t-4),S(t-5)*, but *S(t),S(t-4),S(t-5)* instead. Treating *S(t)* as the new reference frame *R'* in the 3-image algorithm, we can construct the intermediate mosaic of this iteration not at the frame *S(t-2)* but the final mosaic frame *S(t)* directly.

The third and the other iterations over even earlier image frames are processed in the same fashion. The idea of the algorithm is illustrated in Fig. 7. This way, propagation of intermediate mosaic results is no longer necessary, and all the intermediate mosaic results are constructed at the final mosaic frame. Through the iterations over the images up to the very first one a mosaic could be constructed.

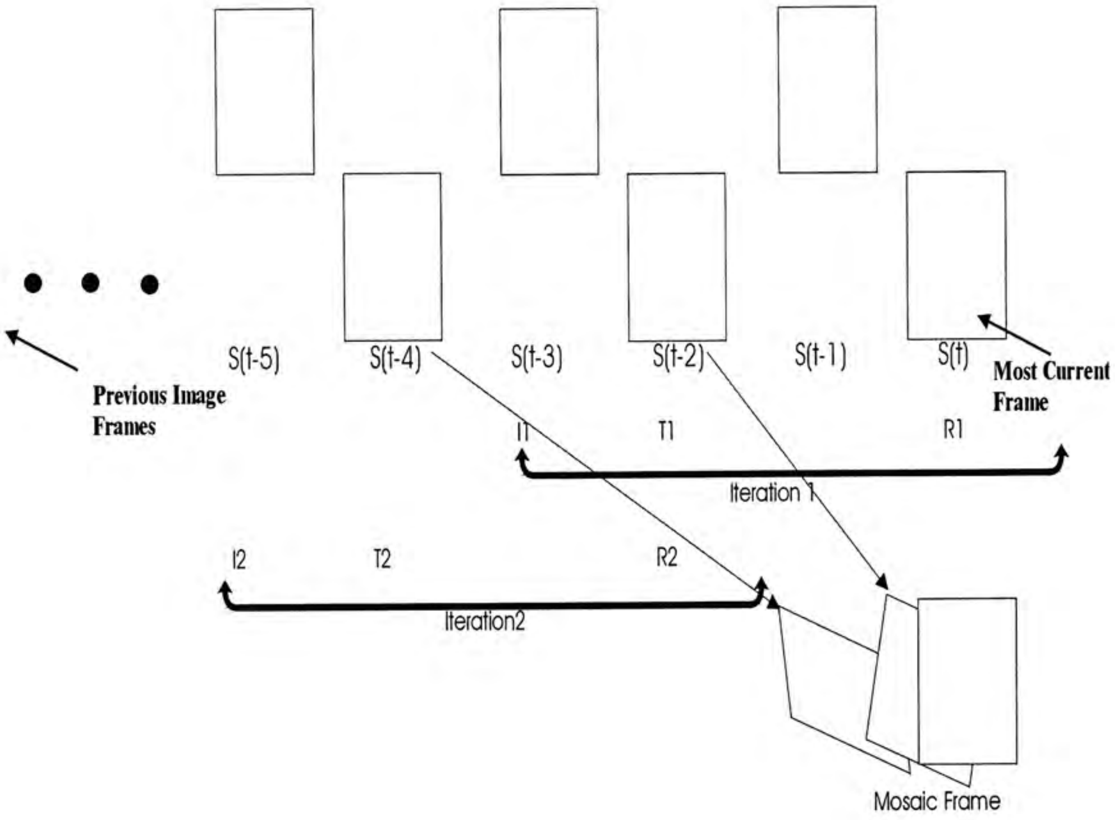


Fig. 7. : Illustration of the n-image algorithm.

Chapter 4. The Uneven-Sampling-Rate n-Image Algorithm

The n-image algorithm employs a fixed-sampling-rate strategy in picking the reference, target, and intermediate frames over the image stream. Even sampling (with respect to time) is not always desirable, as how dense we should have the video stream sampled at a particular section of it should depend upon how close the scene is toward the camera over that particular section. The closer the scene is toward the camera, the faster the visuals move in the image plane, and the denser the sampling should be so that the images to register are still not too different. On the other hand, the sampling should not be so dense that the reference, target, and intermediate images are actually all displaying the same data. Experimental results echo this argument. We found that different sampling interval for the reference, target, and intermediate images in each iteration could result in mosaic of different qualities. In this section we propose methods to allow the above n-image algorithm to have an uneven sampling rate.

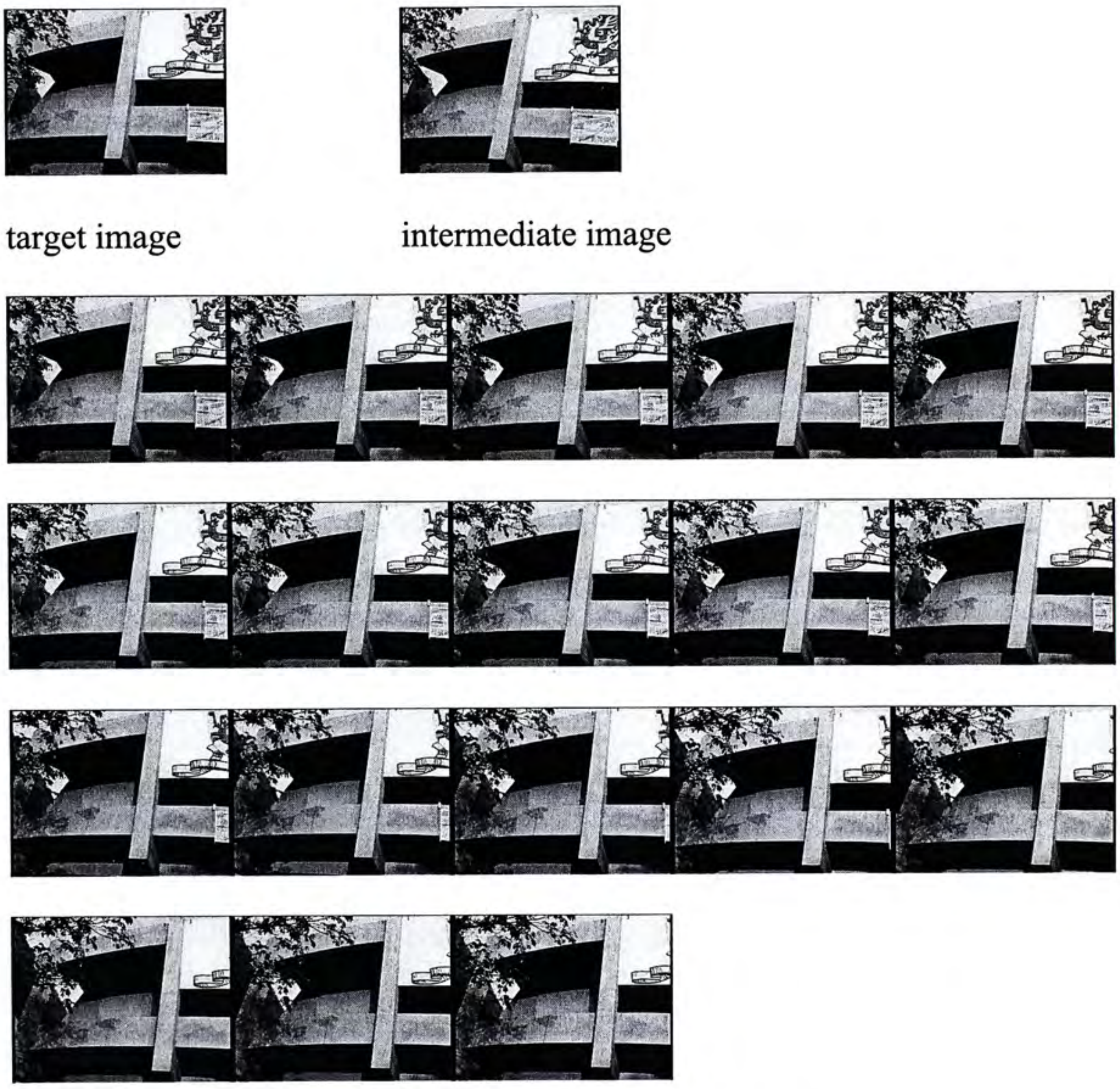
The key is whether we could have a measure of whether the picked images (for the reference, target, and intermediate frames) in any particular iteration are too close or too far apart. We found that the error in projecting the 3D projective coordinates (acquired from the target and intermediate image frames) to the reference image is a good measure.

There are two degrees of freedom in sampling image frames for the three image used in the 3-image algorithm, they are the Reference-Target images separation and the Target-Intermediate images separation. It is obvious that if we change any of the sampling intervals it will affect the error in projecting the 3D projective coordinates to the reference image. So it is the best way for us to vary both the Reference-Target images separation and the Target-Intermediate images separation in order to obtain the best result in the projection of the 3D projective coordinates to the reference image. But it will be an inefficient way in varying both two separations at the same time. So we only want to vary one separation in order to save effort and we have to decide which to be varied. That issue we are going to discuss in following paragraphs.

4.1. Varying the Reference-Target Images Separation

If we fix the separation of the target image and the intermediate image and just changing the separation between the reference image and the target image, that means we are changing the overlapping area of the reference image and the target image. As a result, it will affect how much extra data we will add on the reference image.

Moreover, varying the reference-target images separation also changes the input for estimating the projection of 3D projective coordinates to the reference image (M"). That will affect the accuracy of the projective mapping M". Experimental results show that there is a significance difference in the resulting mosaic when we vary the reference-target images separation. Fig. 8. shows an example of the variation of the projection error to the reference image due to the variation of the reference-target images separation. Fig. 8(a) shows the input images. We fix the separation between the target image and the intermediate image. Then we increase the separation (in image frame) between the target image and the reference image until we cannot obtain matches between the target image and the reference image. The projection error to reference image against the separation between reference image and target image is shown in Fig. 8(b).

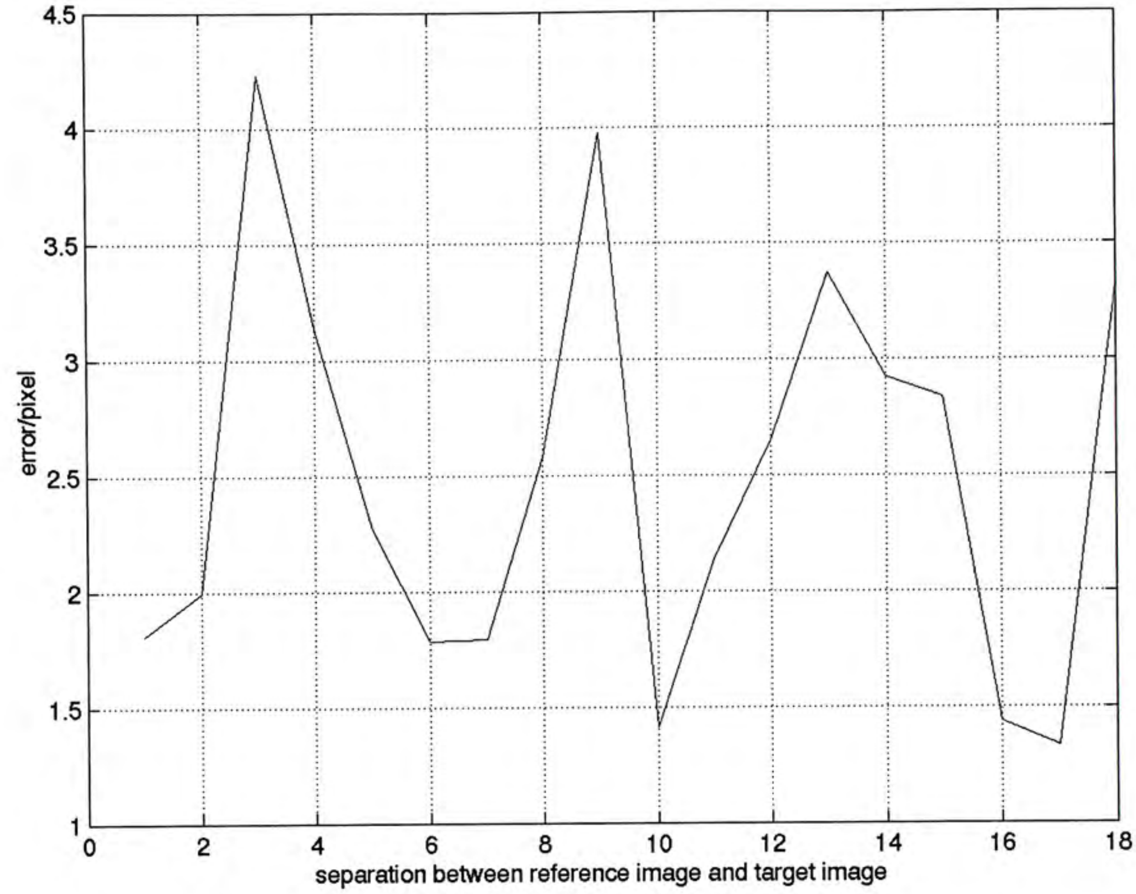


target image

intermediate image

candidate reference images

(a) Input images

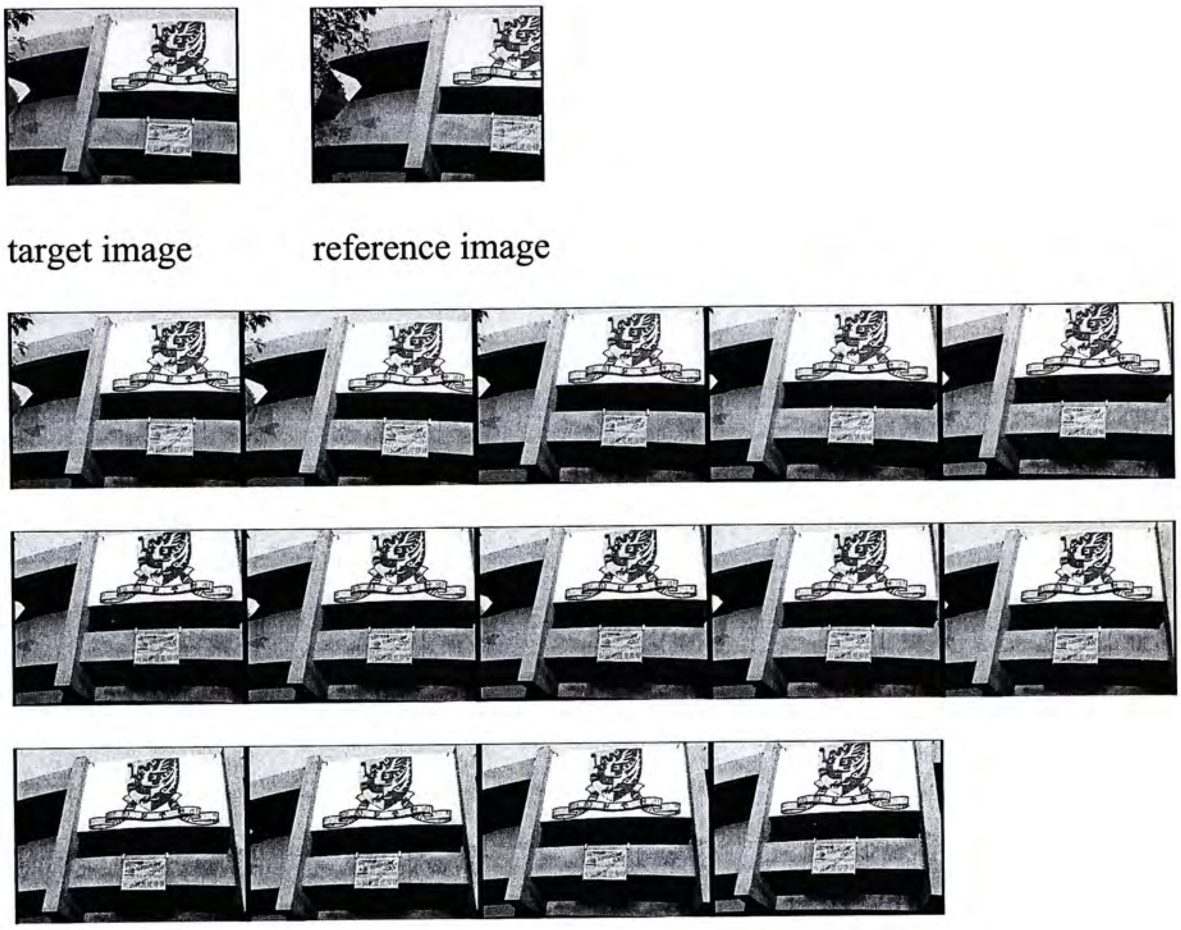


(b) error of projective mapping of reference image against the separation between reference image and target image

Fig. 8. : An example of result of varying the reference-target image separation

4.2. Varying the Target-Intermediate Images Separation

Again, if we fix the separation of the reference image and the target image and just varying the separation between the target image and the intermediate image, that means we are varying the baseline of a stereo reconstruction of a stereo image pair. That also affect the accuracy of the projective mapping M" since we are changing the input for the computation of the projective mapping. Fig. 9. shows an example of variation of the error of projective mapping of the reference image due to the variation of the reference-target images separation. The example is similar to the example in section 4.1. but this time we fix the separation between the reference image and the target image and vary the separation between the target image and the intermediate.

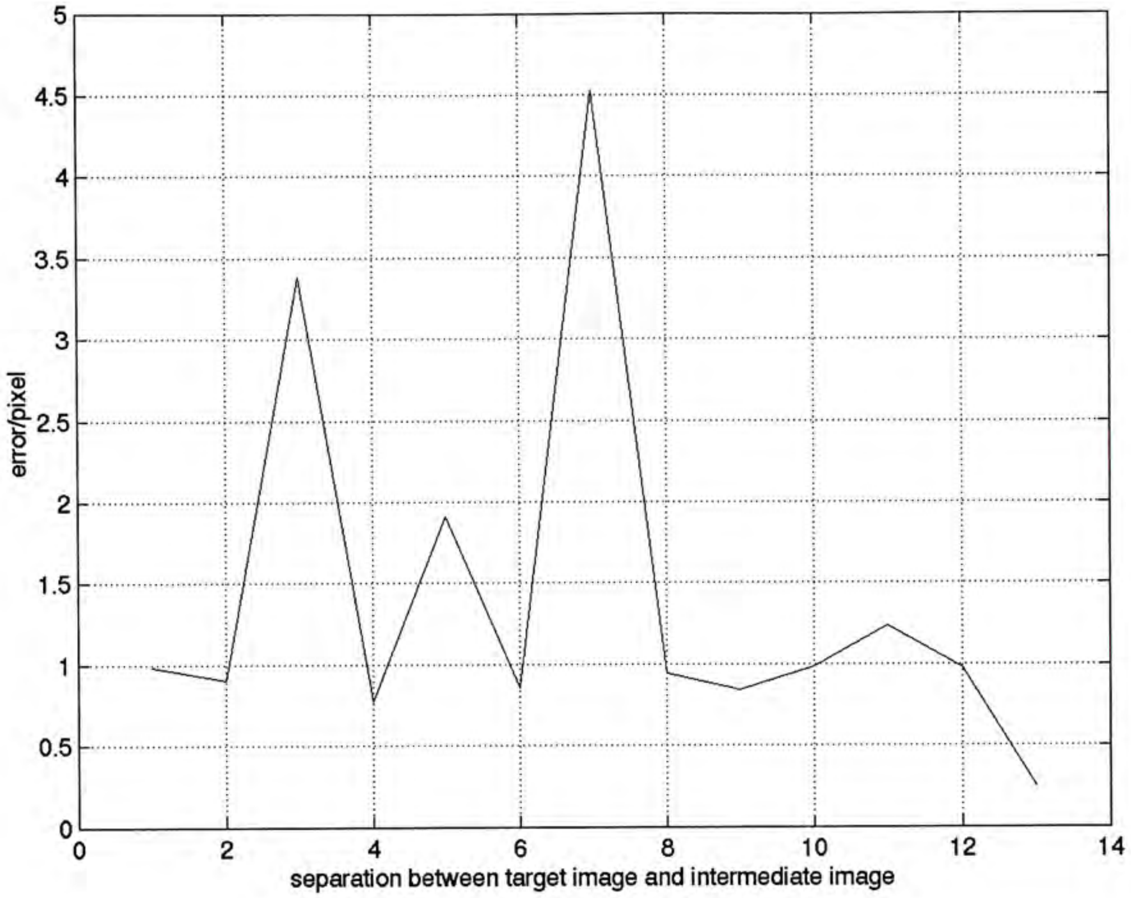


target image

reference image

candidate intermediate images

(a) input images



(b) error of projective mapping of reference image against the separation between target image and intermediate image

Fig. 9. : An example of result of varying the target-intermediate image separation

Besides affecting the accuracy of the projective mapping of the reference image. One thing more important is that the target image and the intermediate image actually is a stereo image pair used for projective reconstruction. Varying the target-intermediate separation means varying the distance of the baseline of a stereo image pair. If we vary the distance of the baseline of a stereo image pair, the accuracy of the reconstruction of the 3D projective coordinates will be greatly affected. In the worst case, we cannot obtain any 3D data if we put the target image and the intermediate image too close.

Since the approach of varying the target-intermediate images separation is more reasonable and suitable for our situation. So it is recommended to vary the separation between target image and intermediate image in the uneven-sampling-rate n-image algorithm. The algorithm will execute as follow:

In the first iteration we still use the frames *S(t)*, *S(t-2)* as the reference image and target image. But for the intermediate image, we need to search for a frame which shares the most suitable separation with the target image (frame *S(t-2)*). The search will start from frame *S(t-3)* to frames earlier in the input sequence. It will continue until we cannot obtain feature correspondence between the candidate intermediate image and the target image frame. In most of the cases, we find that the suitable intermediate image frame is close to the target image so efficiency of the search is not so much a problem.

We decide which frame will be the best intermediate image by examining the distance between (a) the feature positions projected from the target image to the reference image, and (b) the original feature positions in the reference image. The image frame that contributes the least error will be chosen as the intermediate image of the iteration.

In the second iteration, we pick the frame *S(t-2)* as the reference image frame, and *S(t-4)* as the target image. Again we will examine the distance between (a) the feature positions projected from frame *S(t-4)* to frame *S(t-2)*, and (b) the original feature positions in frame *S(t-2)*, in deciding which frame will be the best intermediate image.

The third and the other iterations over even earlier image frames are processed in the same fashion. With decisions about which image frames are to be the target, intermediate, and reference images in each iteration, we could proceed with the n-image algorithm as detailed in the previous section.

Chapter 5. Experiments

5.1. Experimental Setup

The algorithm has been implemented in TargetJr code. TargetJr was a C++ software environment used for computer vision research and image processing applications. The detail of TargetJr can be found in [27]. The task of the system implemented in TargetJr is to perform projective reconstruction and construction of the final mosaic from the matches we obtained from *image-matching*.

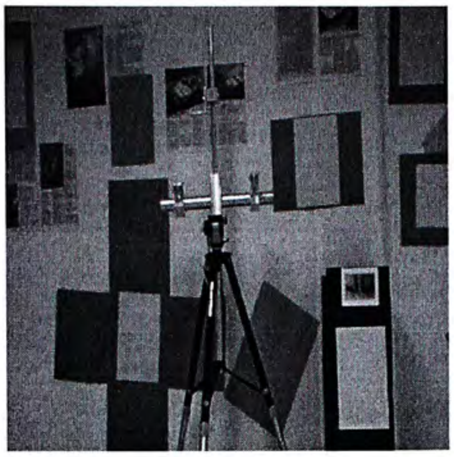
5.1.1. Measuring the Performance

Besides judging the resulting mosaic visually. The performance of the system will be measured with (1) between the feature positions projected from the target image to the reference image, and the original feature positions in the reference image, (2) the time used to construct the resulting mosaic.

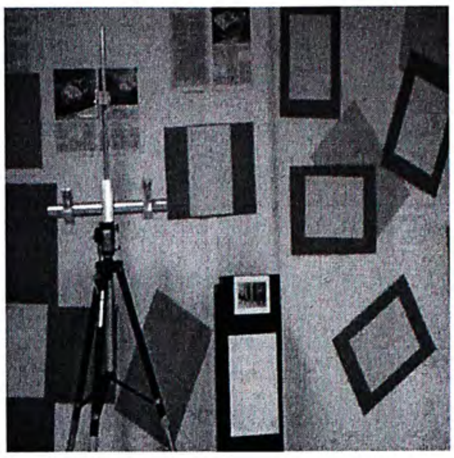
5.2. Experimental on 3-Image Algorithm

5.2.1. Planar Scene

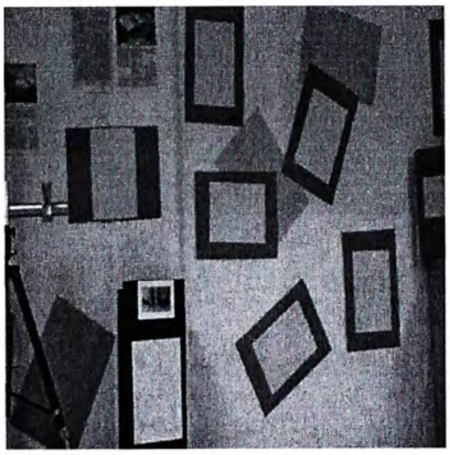
In the first experiment, we want to conduct an initial test on our system of 3-image algorithm. A set of three 512×512 images of an indoor scene taken by the same camera was used. This set of images was downloaded from the public domain at ftp://krakatoa.inria.fr/pub/. As we can see the scene shown in the images is a planar scene. The input images and the resulting mosaic are shown in Fig. 10. The resulting mosaic shows that area that can be seen in the target image but not in the reference image was added to the reference image correctly as we expected. To construct the resulting mosaic in this experiment need 2 minutes 27 seconds. The average projection error to the reference image is 1.0168 pixel over 45 feature points.



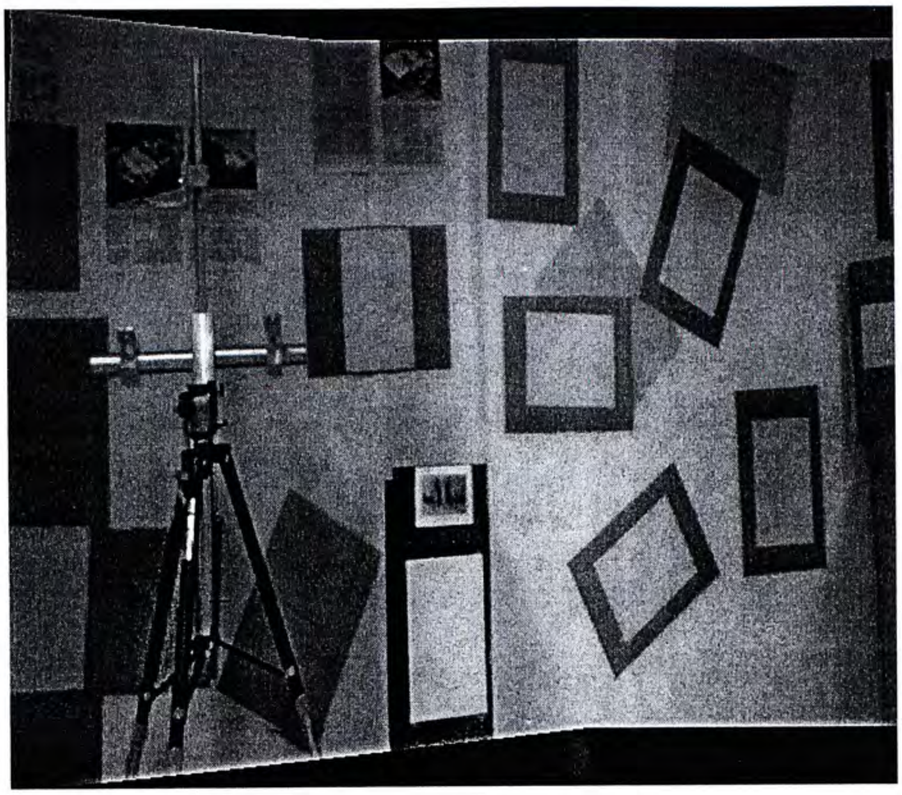
The intermediate image



The target image



The reference image



Mosaic using 3-image algorithm

Fig. 10. : The three images used in the first experiment and the obtained mosaic.

5.2.2. Comparison between a Global Parametric Transformation and the 3-Image Algorithm

After proving that the 3-image algorithm is workable in constructing mosaic, we want to compare the 3-image algorithm to a method that using global parametric transformation. Homography was chosen to be the global parametric transformation method.

In this experiment, synthetic data was generated to compare the 3-image algorithm to the homography method. We first construct a unit sphere in the virtual space, the point feature of the surface of the unit sphere will be project onto three images by a virtual camera with a perspective projection as:

$$\begin{cases} u = f\dfrac{X}{Z} \\ v = f\dfrac{Y}{Z} \end{cases}$$

where (u, v) is the image position of the projected feature point, (X, Y, Z) is the 3D position of the point feature relative to the camera and f is the focal length of the camera which was set to 1. The camera was placed at a position with a distance d from the surface of the unit sphere. The images are the reference image with the optical center on the same line of the center of the sphere. The optical center of the target image had a shift of 0.5 unit in the y-direction to the reference image. The intermediate image shifting 0.2 unit in the y-direction to the target image.

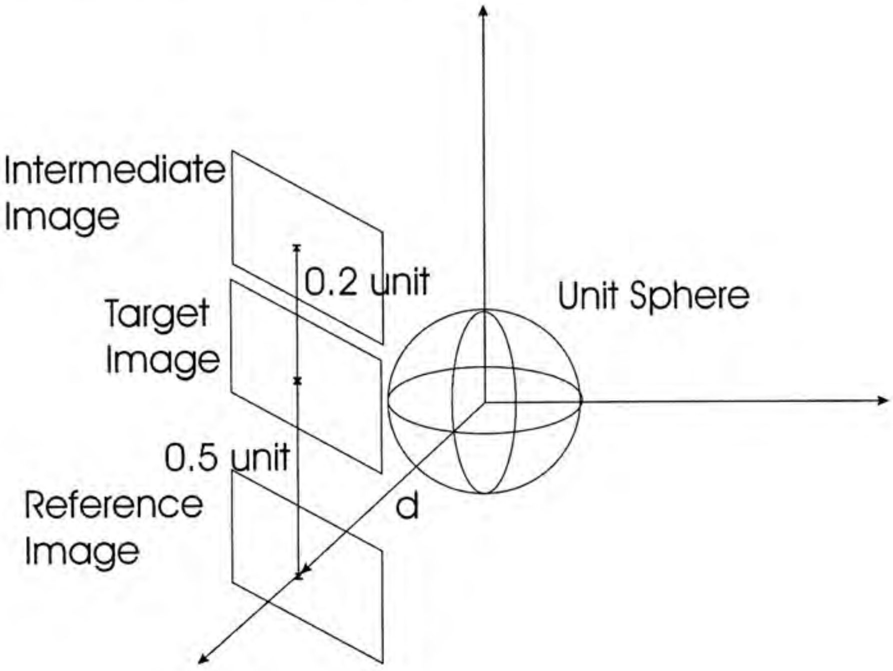


Fig. 11. : Illustration of the experiment.

Next we will calculate the homography of the target image and the reference image, comparing its error to that of the 3-image algorithm on the three images. After that, we will change the distance d between the camera and the sphere to see the change of the error due to the change of the distance between the camera and the scene (Fig. 12). It can be seen that the effect of decreasing distance between the images and scene is less significant for the 3-image algorithm than the homography method.

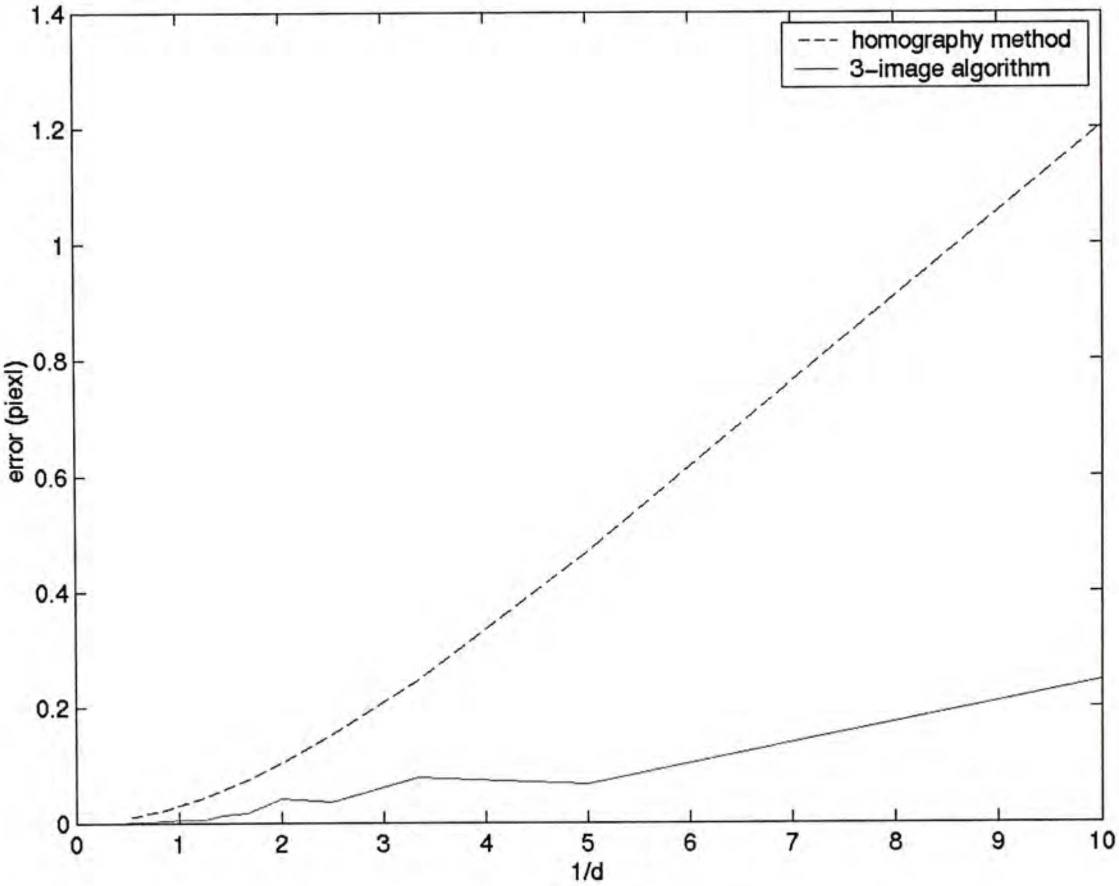
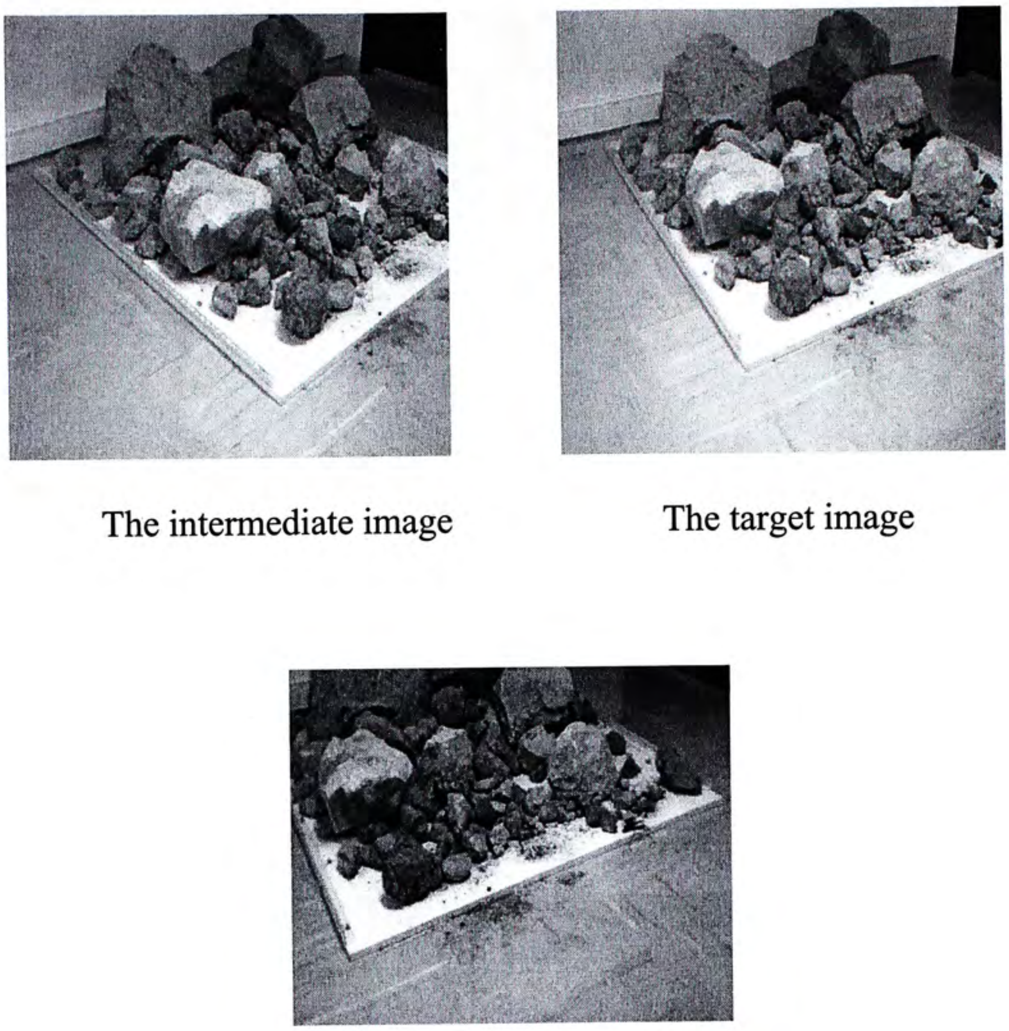


Fig. 12 . Error form homography method and the 3-image algorithm as the distance of the camera from the scene is changing.

5.2.3. Generic Scene

In this experiment, a set of three image of a generic scene was used. This set of images was downloaded from the public domain at *ftp://krakatoa.inria.fr/pub/*. The image size of the target image and intermediate image are 512×512 pixels, and the reference image is of size 512×412 pixels. Both the single homography and multiple homography approaches were applied to the image set in order to find out their difference in constructing mosaic of generic scene using the 3-image algorithm. It can be seen that the mosaicking results obtained by the two approaches are different. Straight lines broke into discontinuous line segments in the merging region of Fig 13(b), and that did not happen in Fig. 13(c).

The projection errors to the reference image are different in the two approaches. The average projection error in single homography approach is 2.94995 pixels over 144 common feature points. When we applied the multiple homography approach to the same set of input images, the average projection error was reduced to 2.0025 pixels over 144 common feature points. It shows that the multiple homography approach is more suitable for a generic scene although it requires more processing time. We used 2 minutes 10 seconds to construct the resulting mosaic shown in Fig. 13(b), 2 hours 10 minutes to build that shown in Fig. 13(c).

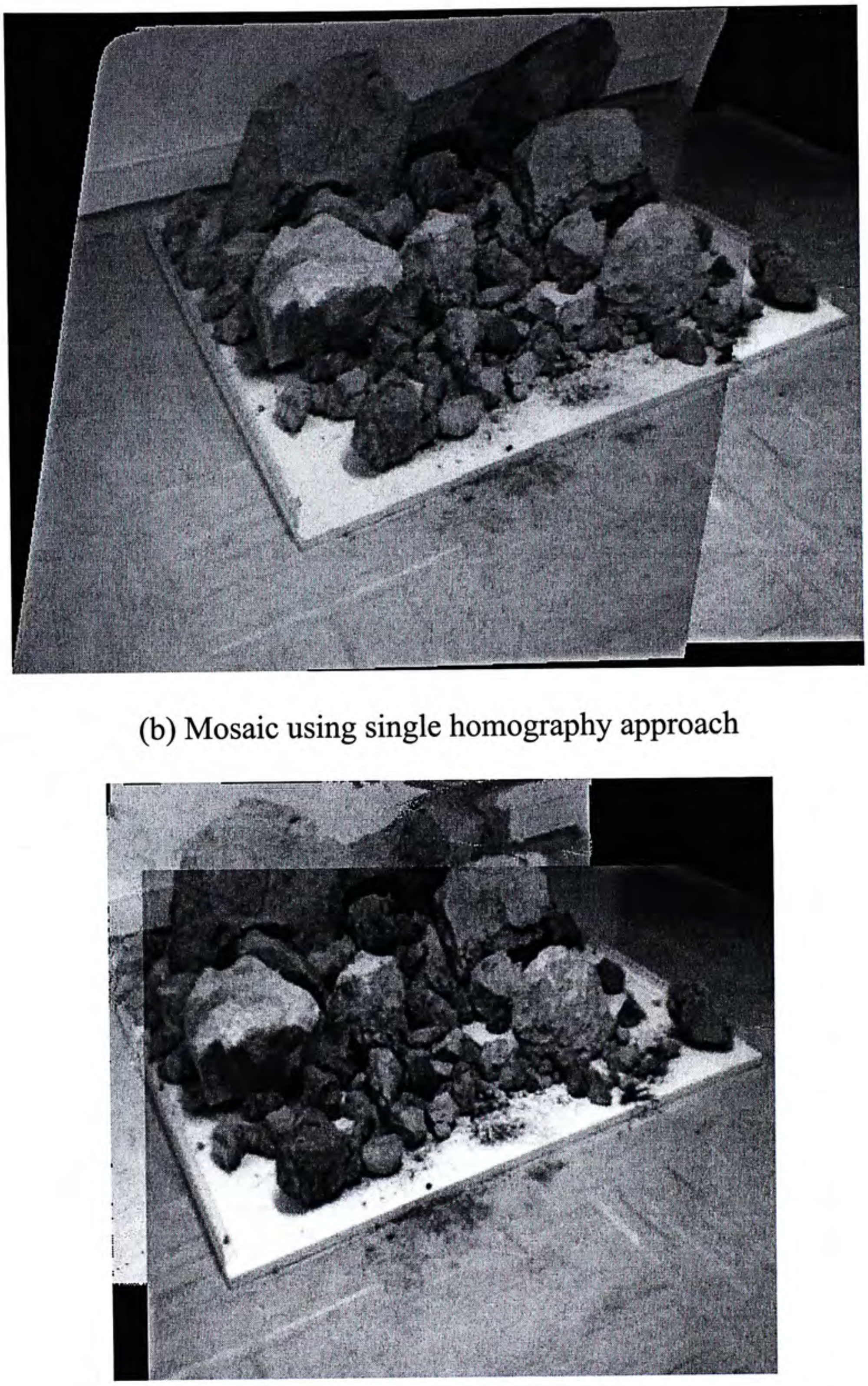


The intermediate image

The target image

The reference image

(a) The input images



(b) Mosaic using single homography approach

(c) Mosaic using multiple homography approach

Fig. 13. : Comparison of the single homography approach and the multiple homography approach.

5.2.4. The Triangular Patches Clustering against the Multiple Homography Approach

Although the multiple homography is workable on some generic scene but later we found that it cannot crop with scene in the real world. When we applied it on a set of images of the CUHK campus, we obtain the result as Fig 14(c). We can find out that many regions are missing in the resulting mosaic, it was because that the closest features used to define reference plane for this regions was ill posed. So regions are projected to somewhere that cannot be seen in the image.

Then we try to apply the triangular patches clustering method on the same set of image. This time we obtain a better result as Fig. 14(d), the resulting mosaic become more presentable. It shows that using the Delaunay triangulation is more suitable than using the multiple homography approach in the images of a generic scene. The average projection errors on the reference image is 7.43332 pixels over 108 common feature points for both approaches. The two approaches having the same value in the error on projected features onto reference image because the error measured here was the error of the projective mapping of the reference image and both approaches occupy the same algorithm in computing the projective mapping. The time used in constructing the mosaicking result shown in Fig. 14(c) is 2 hours 32 minutes. It takes 4 minutes 17 seconds for result shown in Fig. 14(d).

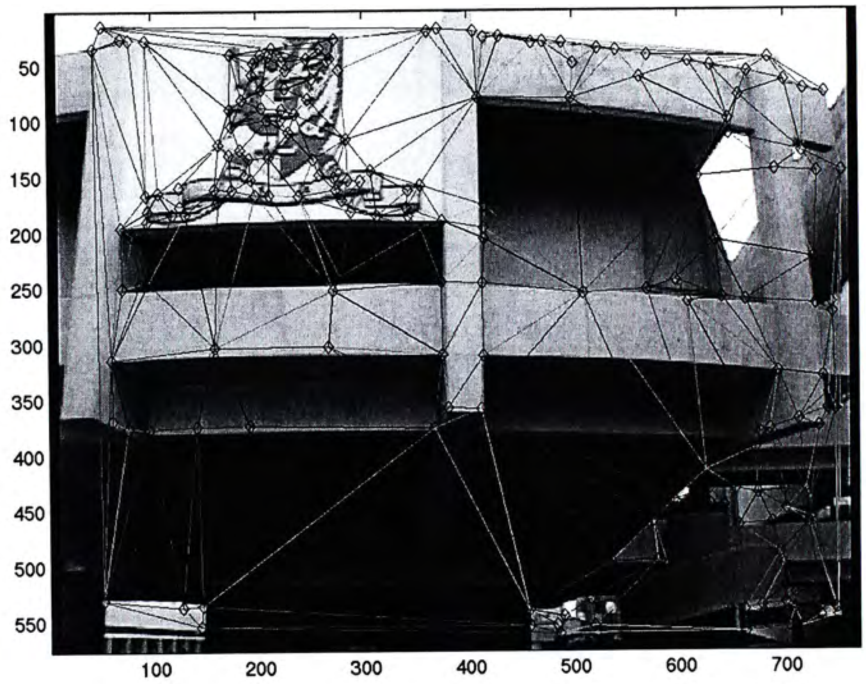
Why does this experiment show a contradicted consequence to the experiment presented in section 5.2.3.. That was because the scene in that experiment was constructed for the purpose of computer vision research so it was well featured and the feature was distributed equally. But for real world, the features were most likely to be distributed randomly so the multiple homography is no longer valid then we have to use another method in this situation.

The intermediate image

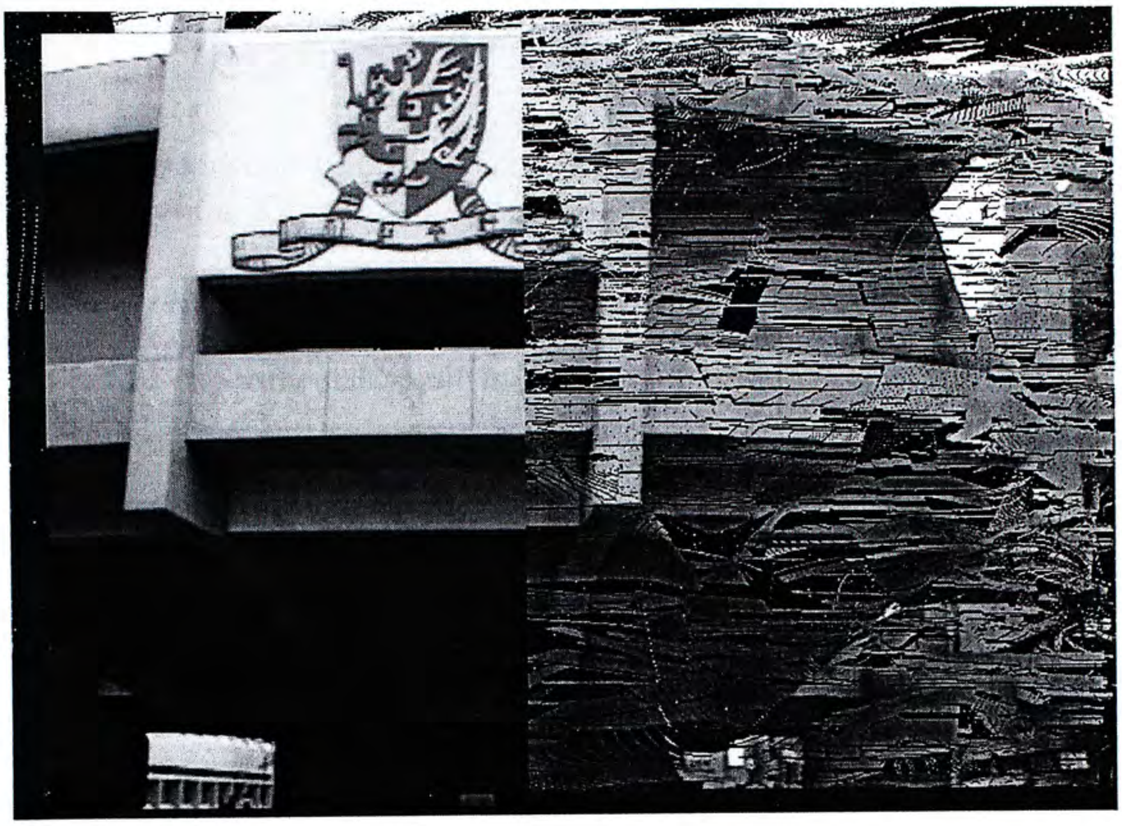
The target image

The reference image

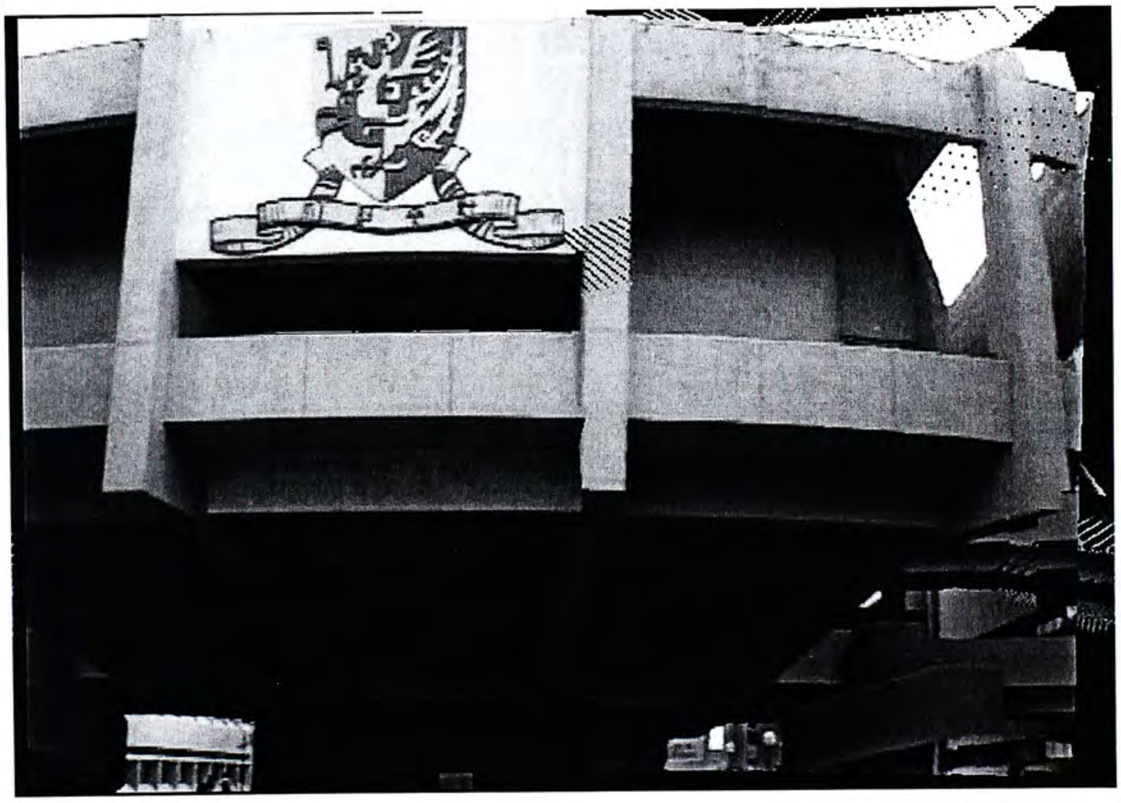
(a) the input images



(b) The Delaunay triangulation of the target image



(c) The resulting mosaic from multiple homography appraoch



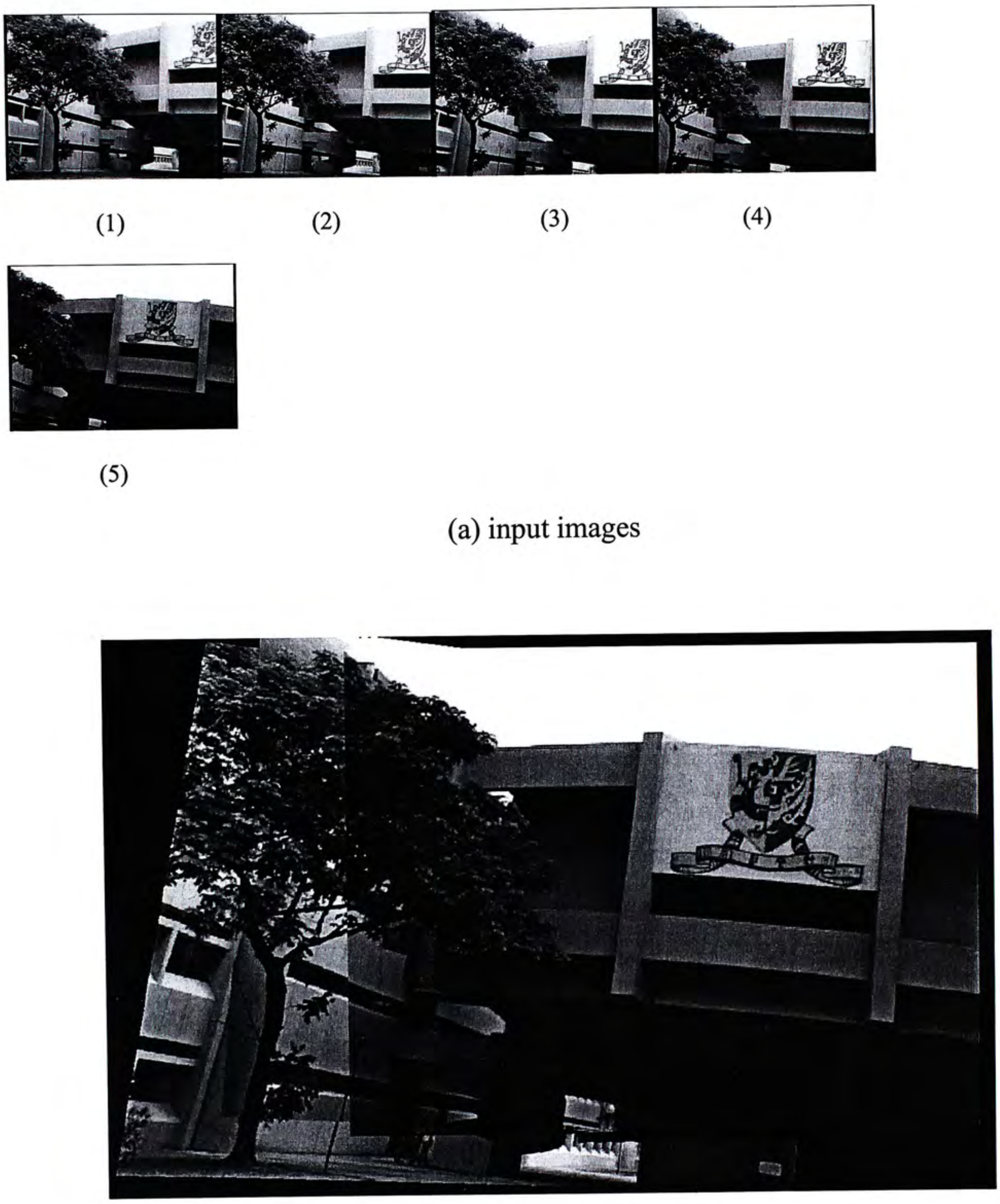
(d)The resulting mosaic from triangular patches clustering

Fig. 14. : Comparison of the multiple homography approach and triangular patches clustering.

5.3. Experiments on the n-Image Algorithm

5.3.1. Initial Experiment on the n-Image Algorithm

The first set of experimental result in n-image algorithm was a video stream of the CUHK campus. Since this is the first test of the implantation of the n-image algorithm so only a sequence of five images was used, that needed two iterations to produce the final mosaic result. The input of image in each iterations is as follow. For iteration one, images (5), (4) and (3) was chosen as the reference image, target image and intermediate image respectively. Then images (4), (2) and (1) become the reference image, target image and intermediate image for the second iteration. The average error in projected features is 0.961888 pixels over 125 common features points for the first iteration, that for the second iteration is 1.04618 pixels over 86 common feature points. The time used to complete the first iteration was 4 minutes 33 seconds and the total time used to build the resulting mosaic was 9 minutes 14 seconds. The result was showed as Fig. 15. It can be seen that the registration is pleasing even the camera motion is a general one.



(1)

(2)

(3)

(4)

(5)

(a) input images

(b) The final mosaic

Fig. 15. : Experimental result of the n-image algorithm toward an image sequence.

5.3.2. Another Experiment on the n-Image Algorithm

Fig. 16 shows another experimental result of the n-image algorithm. We used an image set of seven images (Fig. 16(a)) which was the same set of images as experiment in section 5.2.1. It takes totally three iterations to construct the final mosaic (Fig. 16(c)). The input images for each iterations are as follow (Table 1.).

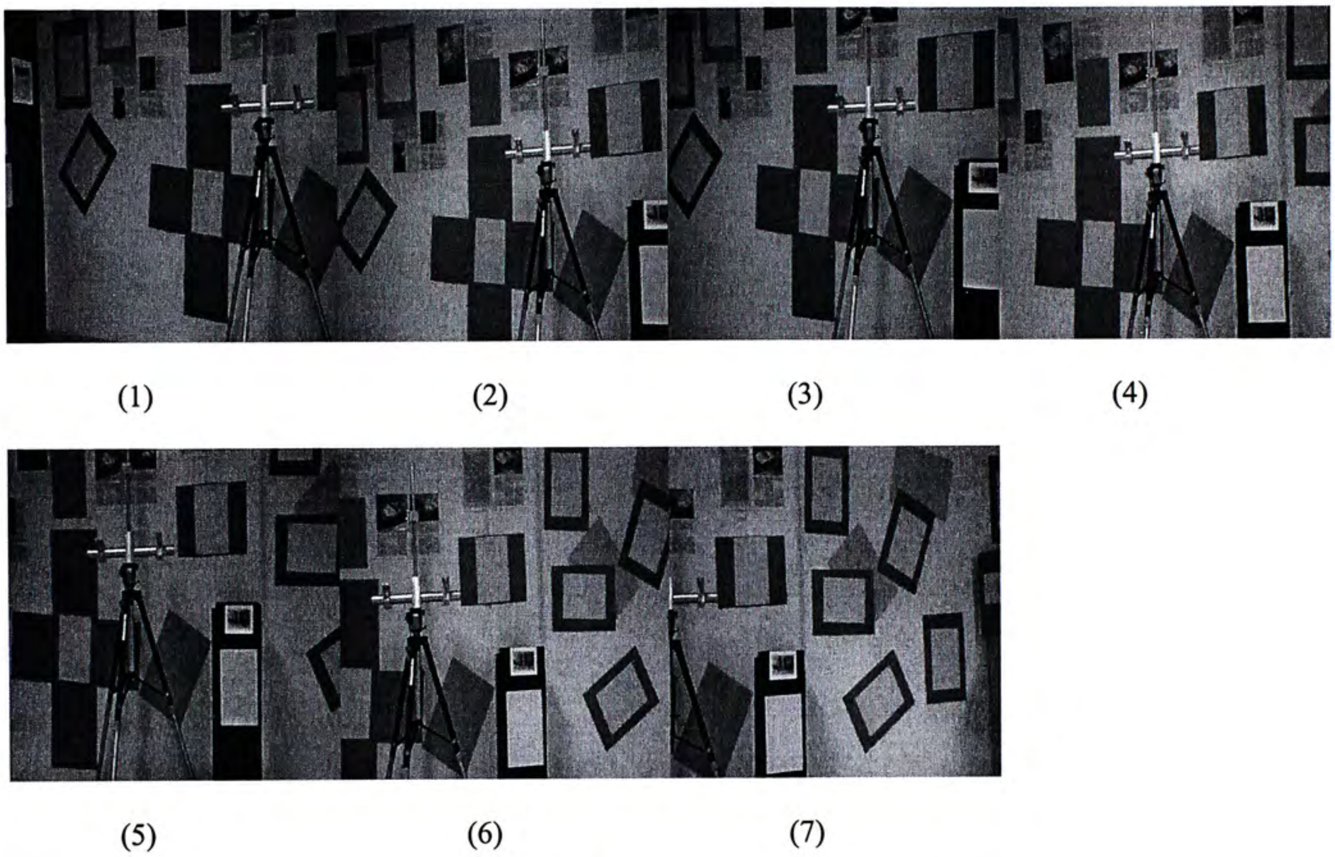
Iteration	Reference image	Target image	Intermediate image
1	(7)	(6)	(5)
2	(6)	(4)	(3)
3	(4)	(2)	(1)

Table 1. The input sequence of images for the system in the experiment in section 5.3.2.

One thing special in this experiment is that we have shown the intermediate mosaic results since they also can be obtained from the system. Fig 16(b) shows the intermediate mosaic result which was obtained after two iterations. The intermediate result obtained after one iteration was omitted in this section since the intermediate mosaic result after the first iteration was same as the result of experiment in section 5.2.1.

We used 2 minutes 27 seconds for the first iteration and the average projection error to the reference image was 1.0168 pixels over 45 common feature points. The second iteration needed 5 minutes 26 seconds to complete and with error of 0.730547 pixels over

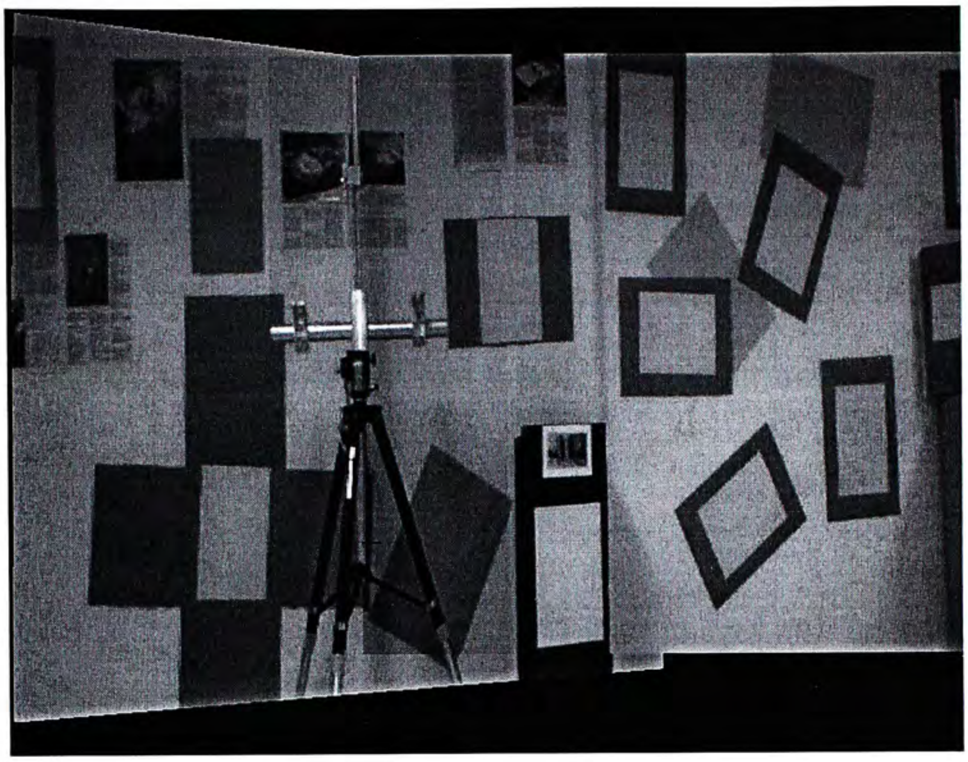
45 common features. The total time used to build the final mosaicking result shown in Fig. 16(c) is 8 minutes and 30 seconds and the error is 1.95044 pixels over 67 common features.



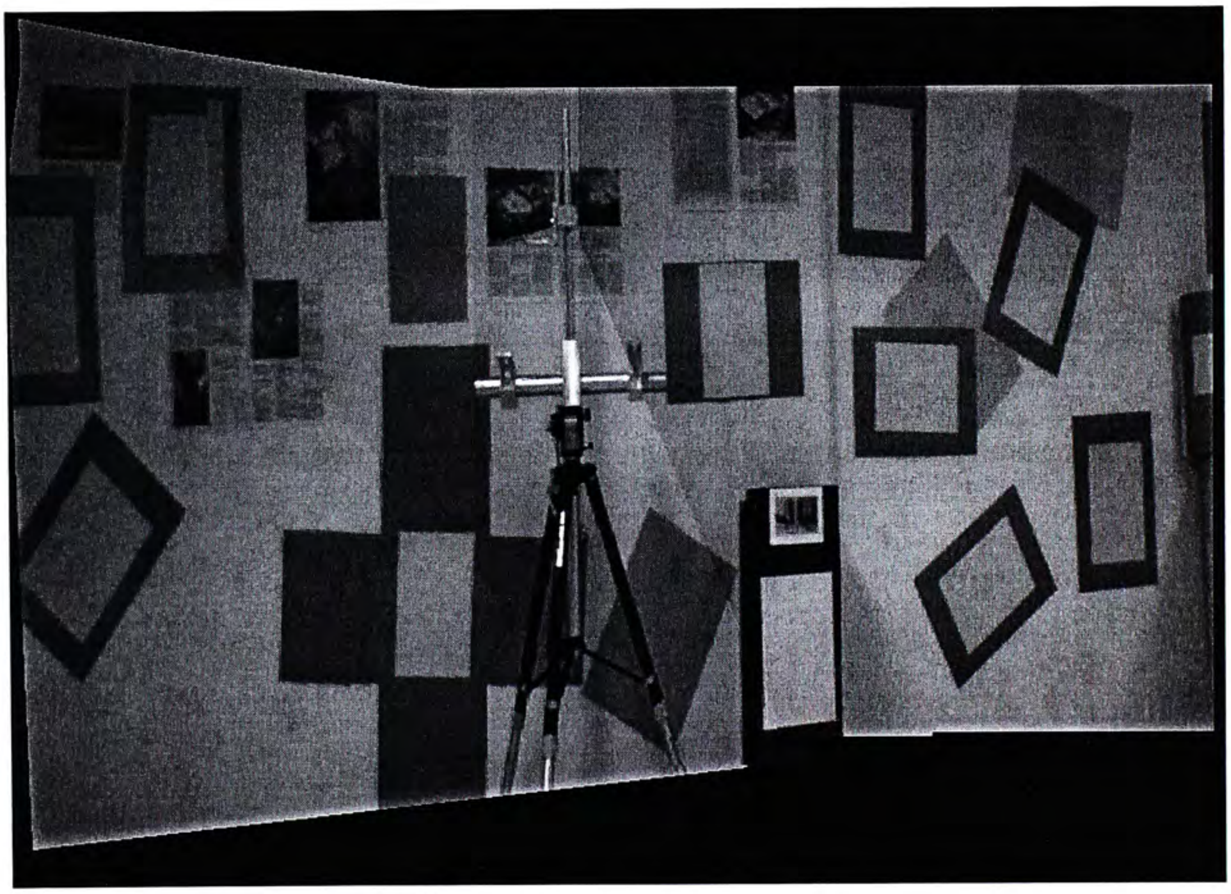
(1) (2) (3) (4)

(5) (6) (7)

(a) Input images



(b) The mosaic after two iterations



(c) The mosaic after three iterations

Fig. 16. : Experimental result of the n-image algorithm with the intermediate mosaic result.

5.3.3. the n-Image Algorithm over a Longer Image Stream

Fig.17 shows another set of results over a longer image-sequence and a scene much closer to the camera. There are totally 13 images in the sequence. The n-image algorithm requires six iterations to construct the final mosaic. The input images for the system in each iterations is as follow (Table 2.).

Iteration	Reference image	Target image	Intermediate image
1	13	12	11
2	12	10	9
3	10	8	7
4	8	6	5
5	6	4	3
6	4	2	1

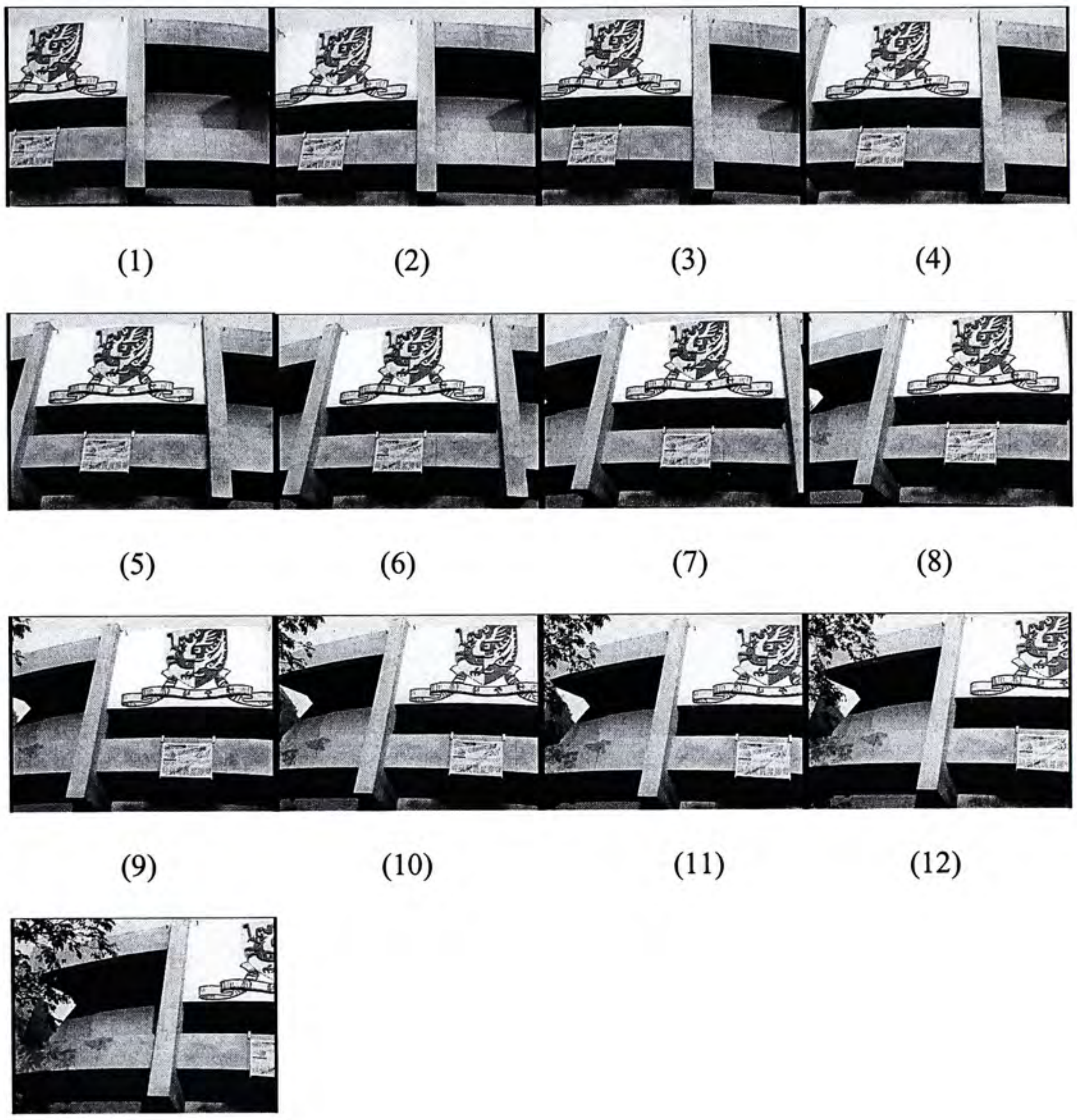
Table 2. The input sequence of images for the system in the experiment in section 5.3.3.

Mis-registration can be seen in the final mosaic result is due to the fixed sampling of the video stream for the reference, target, and intermediate images in each iteration.

Iteration	Projection error	# of common feature	Time used
1	4.27406	80	4’30”
2	1.87319	57	8’57”
3	3.25864	85	13’41”
4	1.87832	114	19’
5	2.18778	110	24’30”
6	5.30416	86	30’56”

Table 3. Performance of the system in the experiment in section 5.3.3.

From the performance showed in Table 3. we find that the propagation of image alignment errors is a major problem for the system.



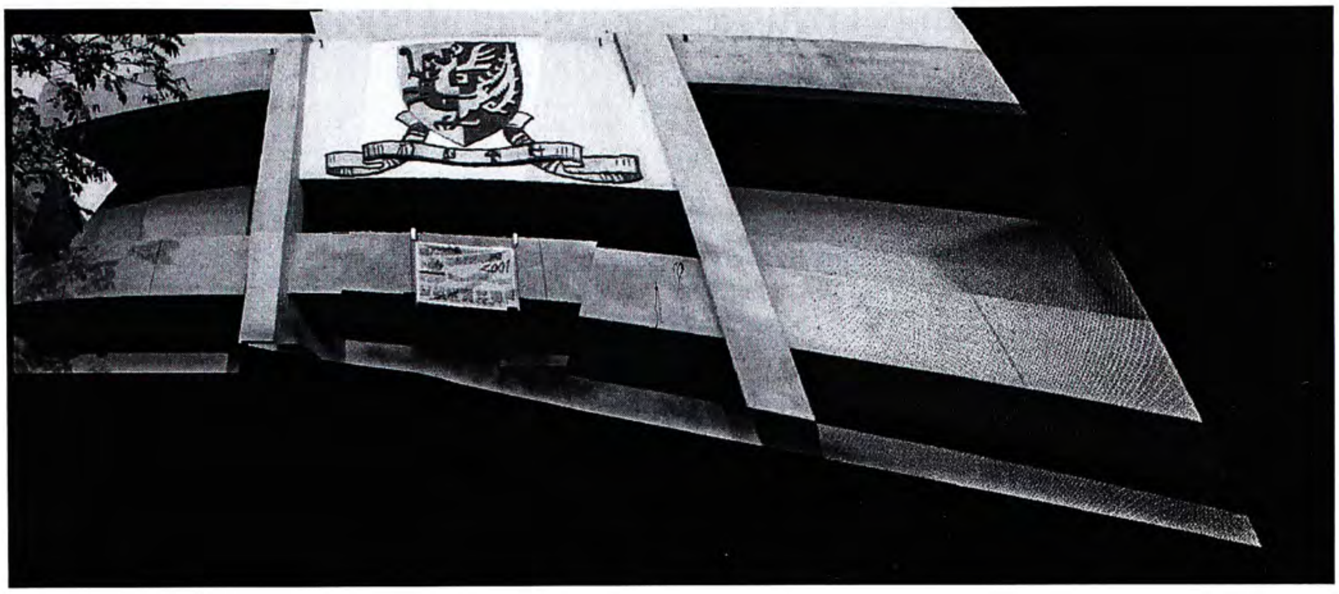
(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11) (12)

(13)

(a) Sampled Input Image Stream



(b) The final mosaic

Fig. 17. : Experimental result of the n-image image algorithm over a longer image stream and a "closer" scene.

5.4. Experiments on the Uneven-Sampling-Rate n-Image Algorithm

5.4.1. Varying Reference-Target Images Separation

After obtaining the result in experiment presented in section 5.3.3., modification have be made on the n-image algorithm. Uneven-sampling-rate was use instead of the fixed-sampling-rate method.

Fig. 18 shows the experimental result of using the varying reference-target images separation method to sample the input images. There are totally 13 images in the image sequence. It takes six iterations to construct the final mosaic result. The use of images for each iteration is as follow (Table 4.).

Iteration	Reference image	Target image	Intermediate image
1	13	12	11
2	12	10	9
3	10	8	7
4	8	6	5
5	6	4	3
6	4	2	1

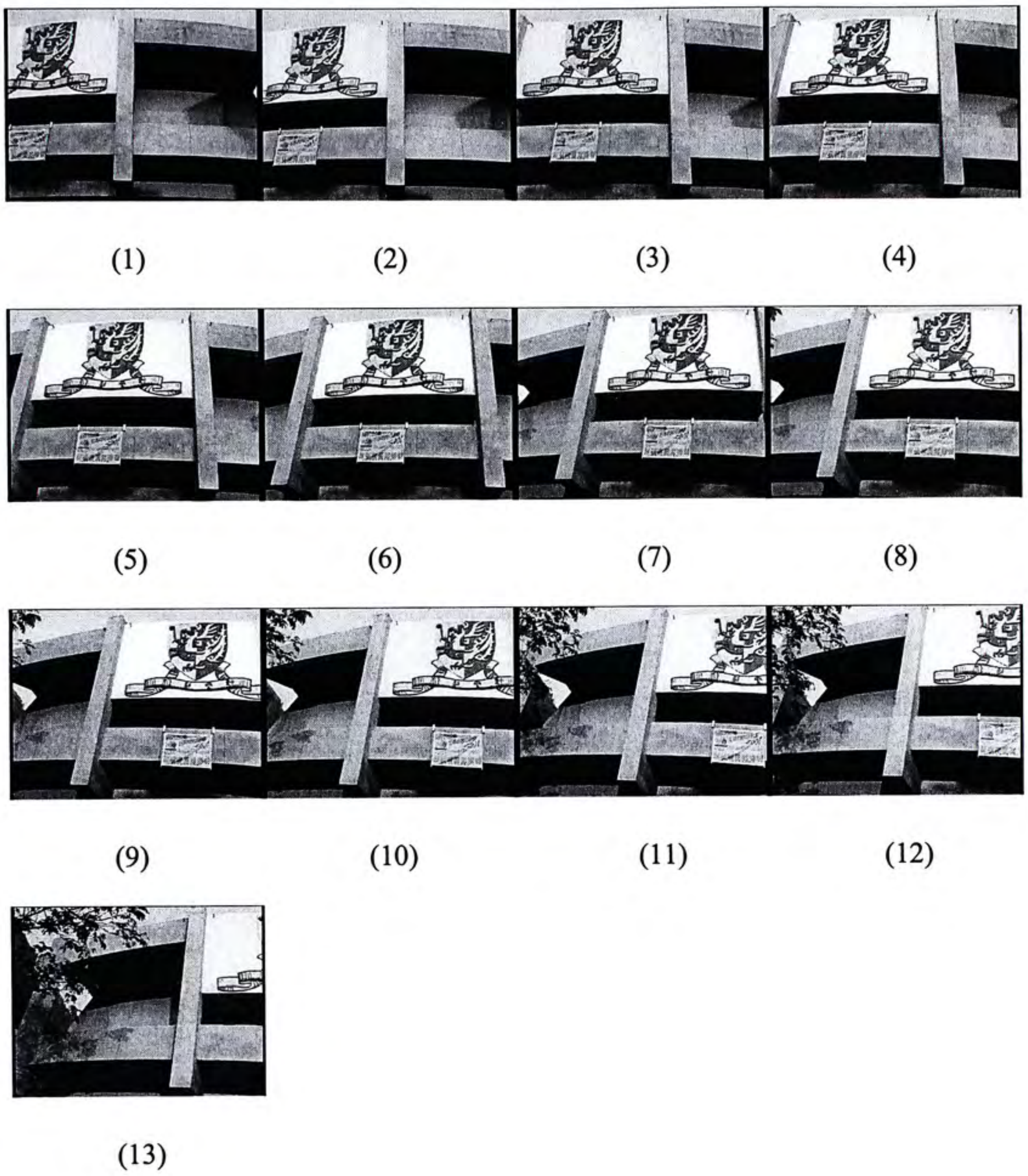
Table 4. The input sequence of images for the system in the experiment in section 5.4.1.

The image sequence contains same number of image as that of experiment in section 5.3.3. but the final results show there are significant difference between the two approaches. It can be seen that the final mosaic result (Fig. 18(b)) is much pleasing when compare with the mosaic result of the experiment (Fig. 17(b)). One should notice that the sampled image input is different to that of the experiment 5.3.3 since they use different methodology in sampling the input images. Even though the images were captured from the same video stream. We can find out the difference in the error of projected features which was as follow (Table 5.).

Iteration	Projection error	# of common feature	Time used
1	1.43804	26	4'22"
2	0.787325	75	8'50"
3	0.822752	95	13'50"
4	0.701881	72	18'55"
5	1.42258	98	24'22"
6	2.05888	61	30'30"

Table 5. Performance of the system in the experiment in section 5.4.1.

It can be seen that the growth of error for this approach is less rapid than that of the approach with even sampling. The time used for both approaches are almost the same because the time used for computing depends on the number of iteration but not the sampling technique.



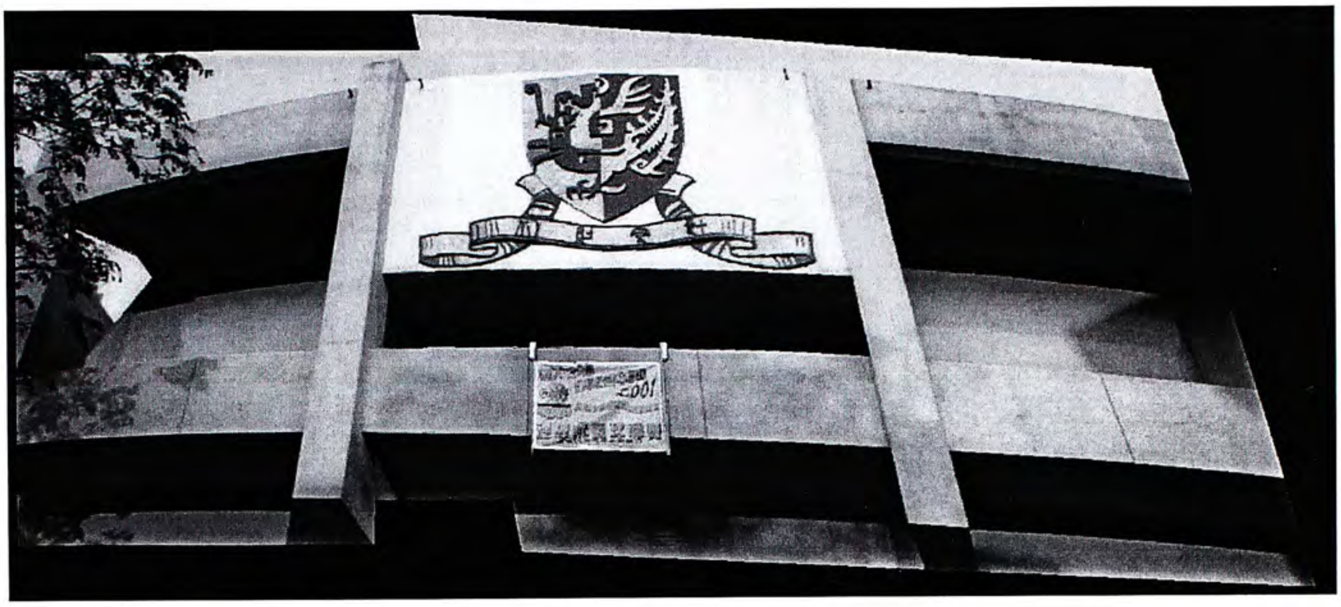
(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11) (12)

(13)

(a) Sampled input images



(b) The final mosaic

Fig. 18. : Experimental result of the uneven-sampling n-image algorithm using the varying Reference-Target separation method.

5.4.2. Varying Target-Intermediate Imges Separation

Fig. 19 shows the experimental result of using the varying target-intermediate images separation method to sample the input images. There are totally 17 images in the image sequence. They were captured from the same video stream as the experiments in section 5.3.3. and section 5.4.1. It takes eight iterations to construct the final mosaic result since the sequence contains more images than experiment in section 5.4.1. The use of input images is as follow (Table 6.).

Iteration	Reference image	Target image	Intermediate image
1	17	16	15
2	16	14	13
3	14	12	11
4	12	10	9
5	10	8	7
6	8	6	5
7	6	4	3
8	4	2	1

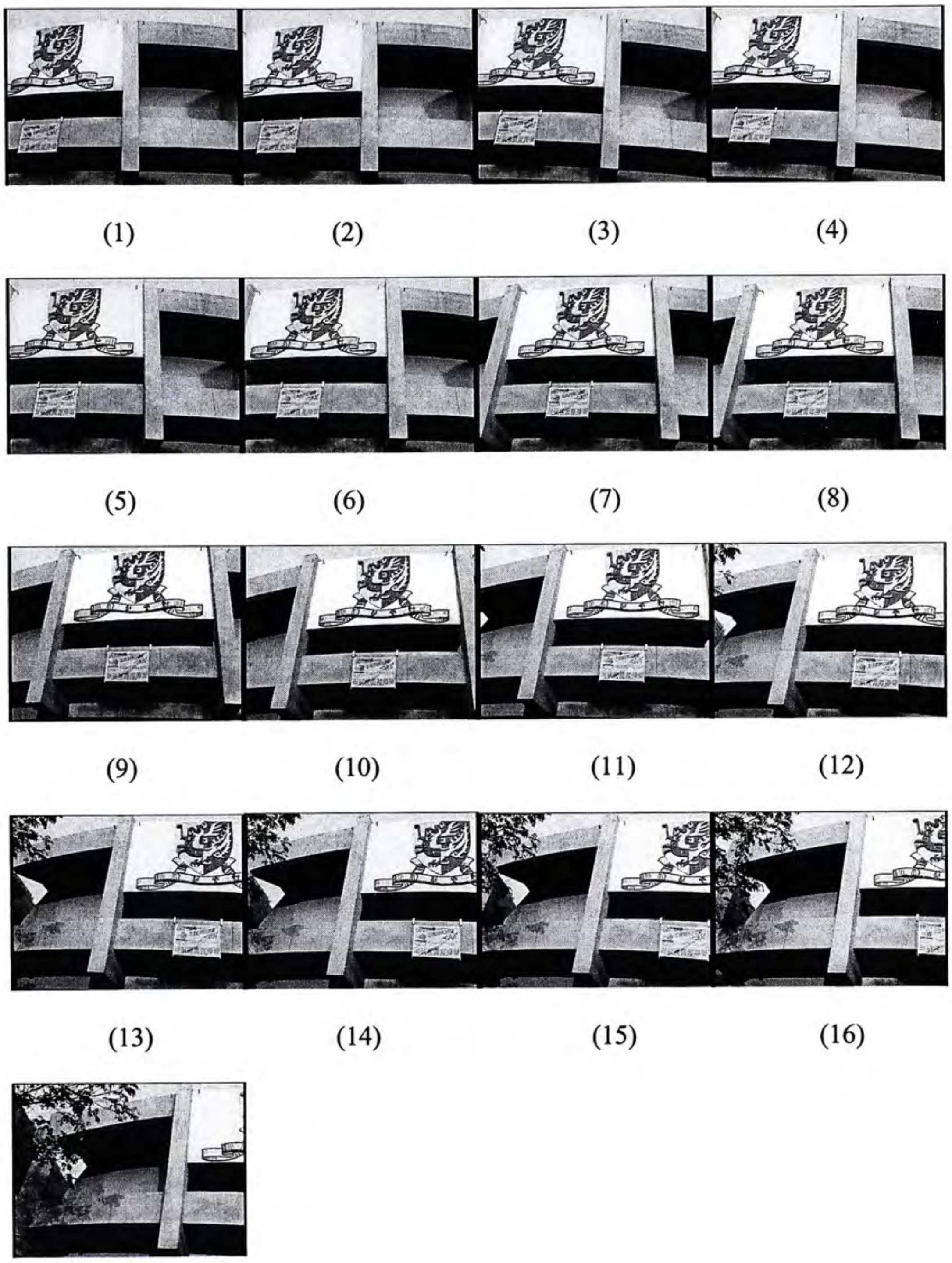
Table 6. The input sequence of images for the system in the experiment in section 5.4.2.

The final mosaic result was as pleasing as that of experiment in section 5.4.1. That means the both sampling technique are workable to be applied as tool of uneven sampling. The average error in projected features and time used to construct the mosaic is shown in Table 7.

Iteration	Projection error	# of common feature	Time used
1	1.33597	41	4'40"
2	1.2546	61	9'12"
3	1.05658	57	14'05"
4	0.925117	102	19"
5	0.671268	105	24'40"
6	0.983246	102	30'05"
7	1.38879	103	36'
8	1.93668	102	42'

Table 7. Performance of the system in the experiment in section 5.4.2.

It can be shown that varying the target-intermediate separation also reduced the growth of the error of projected features of the reference image. But due to the argument in section 4, varying the target-intermediate separation method is recommended to be use as the tool for sampling input images.



(1) (2) (3) (4)

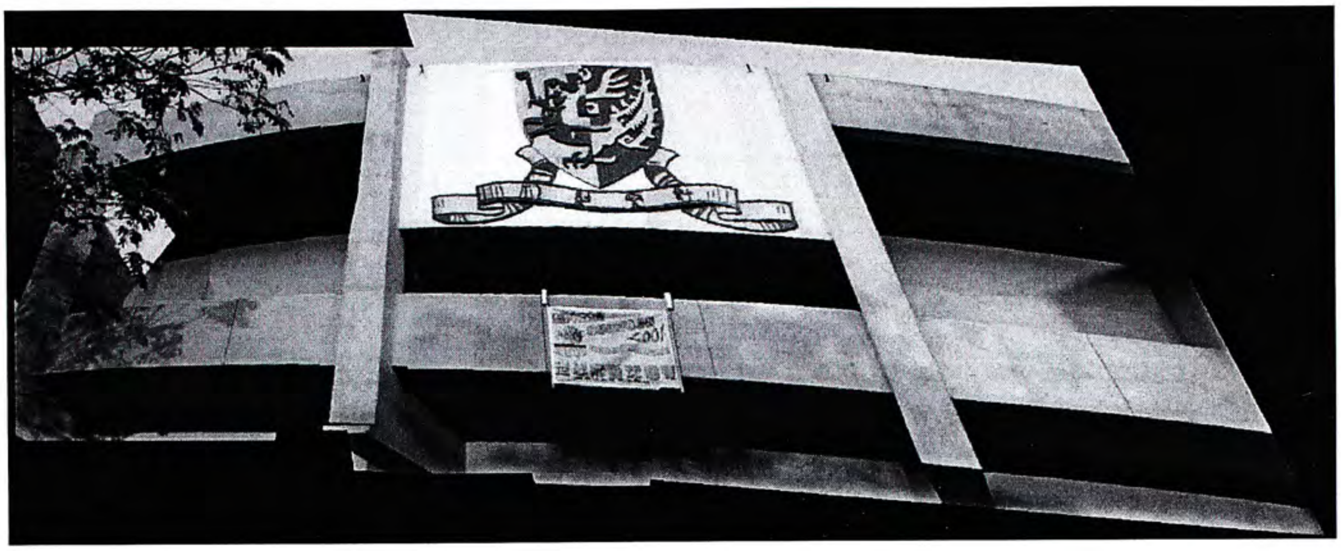
(5) (6) (7) (8)

(9) (10) (11) (12)

(13) (14) (15) (16)

(17)

(a) Sampled Input Stream



(b) The final mosaic

Fig. 19. : Experimental result of the uneven-sampling n-image algorithm using the Varying Target-Intermediate Separation method.

5.4.3. Comparing the Uneven-Sampling-Rate n-Image Algorithm and Global Transformation Method

We are going to present two mosaicking results obtained by difference methods. One is obtained from our algorithm another is obtained by the homography method. The aim of this experiment was to compare the performance of the two approaches.

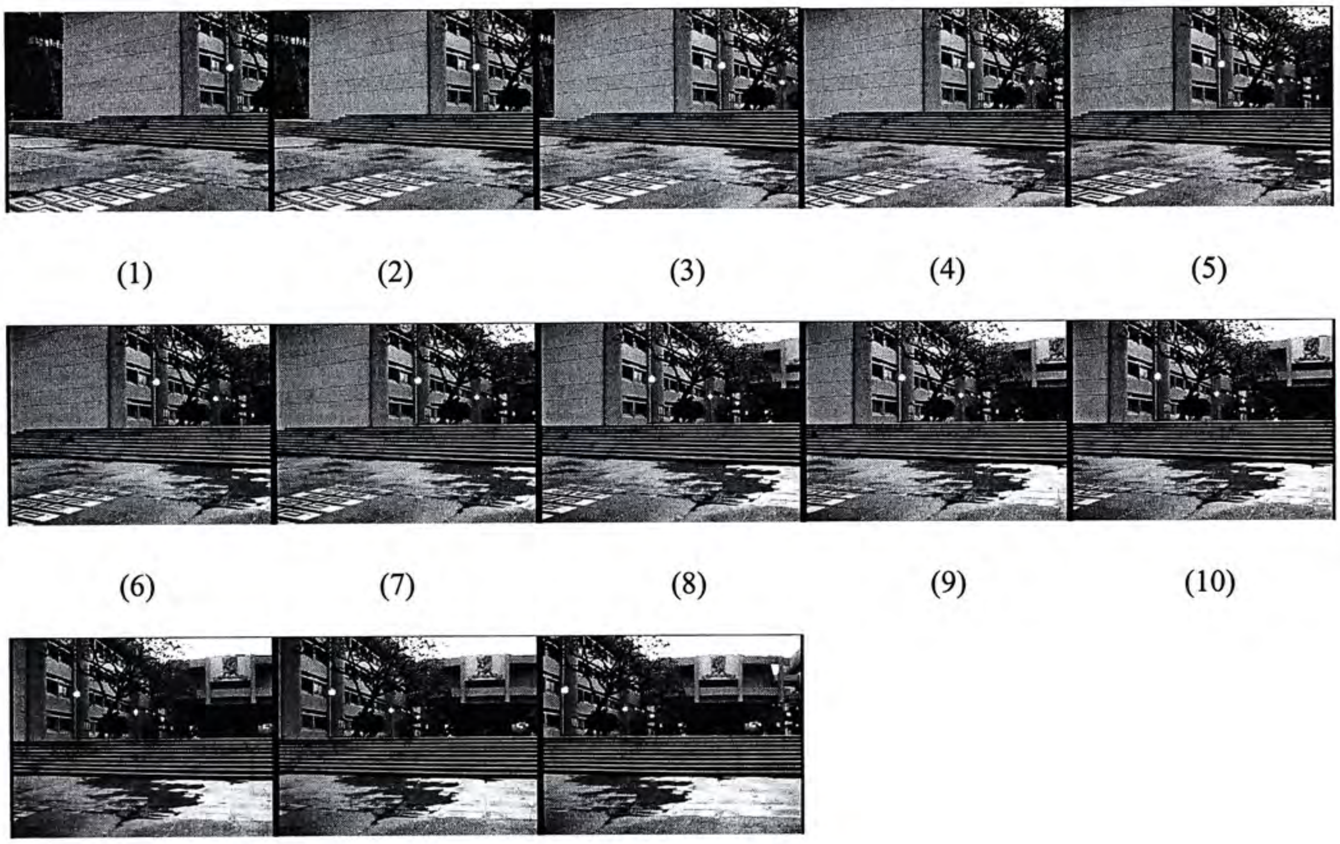
Fig. 20(a) shows the input images captured from a video. The use of images in each iterations in our algorithm have been shown in Table 8.

Iteration	Reference image	Target image	Intermediate image
1	13	12	11
2	12	10	9
3	10	8	7
4	8	6	5
5	6	4	3
6	4	2	1

Table 8. The input sequence of images for the system in the experiment in section 5.4.3.

We get the mosaicking result shown in Fig. 20(b) after a process of six iterations. For the homography method, the same set of input images is used and the result is shown in Fig. 20(c).

When comparing the two mosaics, mis-alignment is very serious in the part of the steps in Fig. 20(c) but that was less obvious in Fig. 20(b). The mis-alignment in Fig. 20(c) was due to the change of distance between the objects in the scene and the camera. But the variation of distance between objects and camera plays a less important role in Uneven-Sampling-Rate n-Image Algorithm. This result agrees with the result of the simulation of section 5.2.2.

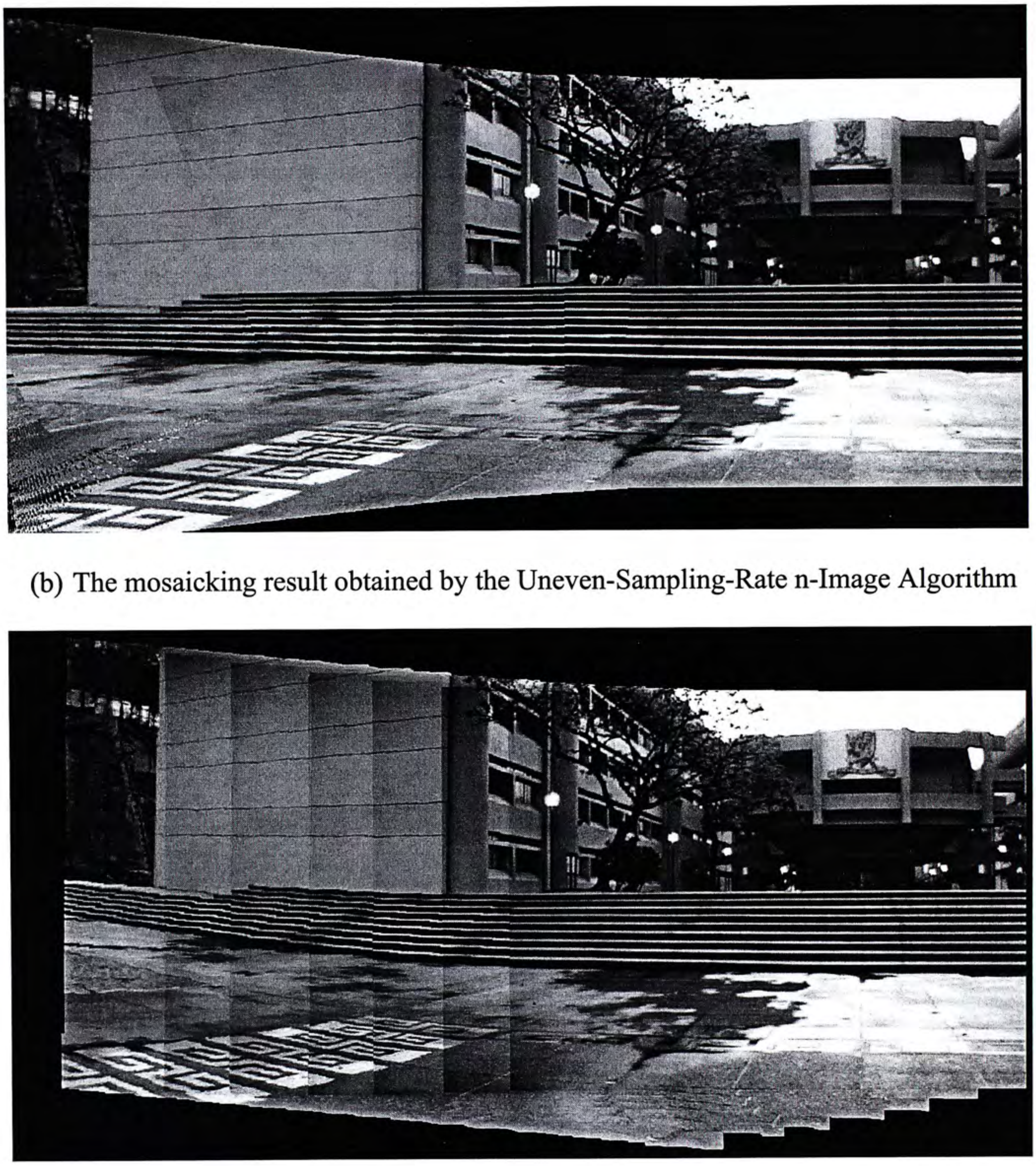


(1) (2) (3) (4) (5)

(6) (7) (8) (9) (10)

(11) (12) (13)

(a) Input images



(b) The mosaicking result obtained by the Uneven-Sampling-Rate n-Image Algorithm

(c) The mosaicking result obtained by homography method

Fig. 20. : Comparison on uneven-sampling-rate n-image algorithm and the homography method

Chapter 6. Conclusion and Discussion

The study and development of a mechanism that allows image mosaicking to be constructed from image data with parallax was reported in this thesis. The further development of the mechanism from the case of a discrete image set to the case of an image stream was also presented.

Parallax in the image data, in the case of an arbitrary scene pictured under arbitrary camera motion, is a challenge to image mosaicking since it causes the absence of the global transformation between the images. This work presents a framework that allows images with parallax to be stitched together and form a mosaic of acceptable quality.

The 3-image algorithm (first proposed by our research group in [3]) uses a third image (the intermediate image) to overcome the problem of parallax in the image data. The algorithm first performs a projective reconstruction to recover the projective coordinates of the scene using the common features between the intermediate image and the target image (the image we like to register) with the reference (the desired mosaic frame). It then re-projects 3D projective coordinates onto the reference image to obtain a mosaic result. The 3-image algorithm does not suffer from mis-registration arisen from the presence of parallax since it does not assume a global parametric transformation between the images. In this work, modifications have been made on the 3-image algorithm to make it more robust and reliable.

With the 3-image algorithm as a basis, we seek to construct mosaic from video stream that contains parallax. The n-image algorithm was proposed to accomplish the aim. The n-image algorithm is an extension of the 3-image algorithm; it has the mechanisms to (1) divide image stream into 3-image sets for the 3-image algorithm to iterate upon; and (2) accumulate intermediate mosaic results over various 3-image sets to compose the final mosaic result.

Experiments show that the fixed-sampling-rate n-image algorithm is insufficient in constructing mosaic from a longer image sequence and for scene that is close to the camera. Erroneous results were obtained in the above situation. The uneven-sampling-rate n-image algorithm was developed to overcome that problem. The key of the uneven-sampling n-image algorithm is to choose the suitable image frames to be the 3-image sets in each iteration. It then uses the image sets to perform the original n-image algorithm. The error in the projecting the projective coordinate to the mosaic frame is used as a measurement in choosing the image frames.

The advantages of the developed algorithms are that the concept is simple, and the algorithms are easy to implement and reliable. Experiment results show that even for images with parallax the final mosaic is of acceptable quality. The results show that the developed algorithms are efficient and reliable for registering images and image stream with parallax. Comparisons have also been made to show the difference in the performance of the methods using global transformation and that of the developed algorithms.

Although we can obtain mosaic of acceptable quality from the developed algorithm but the accumulation of image alignment errors is a major problem for the algorithm. The accumulation errors are due to the structure of the algorithm. For the target image in the iteration *m* of a series of input image, how that target image is transforms to the final mosaic is determined by the transformation of the previous iteration (iteration *m-1*). Error in the result of iteration *m-1* will cause error in the referencing coordinates of iteration *m*. That means we try to transform the target image in iteration *m* to the reference image with an incorrect position.

And error in the result of iteration *m* will cause error in the referencing coordinates of the following iteration (iteration *m+1*). As the mosaicking process continues, the error will accumulate and affect the quality of the mosaicking result.

Even though we have tried to reduce the accumulation of image alignment errors by using the uneven-sampling technique on the input images, it still exists in the process of the developed algorithm. So further research should be done on the existing algorithm to reduce the accumulation of image alignment errors.

Improvements could also be made on the developed algorithms to reduce the error of the transformation mappings used in the algorithm (they include the fundamental matrix, projective mapping and homography) in order to make the algorithms more robust and reliable.

The processing time of the developed algorithm is also a limitation. Currently it takes three to four minutes to complete one iteration and it is too long compared to the processing time of other mosaicking systems. So improvements should be done on the developed algorithm to reduce the processing time.

The triangular patches clustering method used in constructing mosaic might not be as reliable as the single homography method when the scene is a planar or distant. This is because in the method only three points are used to determine a (local) plane in 3D space and it is sensitive to noise in the feature correspondences. But for the single homography method, it assumes a single plane for the entire scene (which is a valid assumption in this particular case), and all feature correspondences are used to determine just a plane in 3D space. The effect of noise in feature correspondence is thus reduced in this case. However, our system has a threshold measurement (please refer section 2.5) to decide which method should be used.

Moreover, the quality of the resulting mosaic is affected by the performance of the feature matching software also is a limitation of the developed algorithm. The developed algorithm requires a quite dense of feature correspondences in the images in order to construct a mosaic with acceptable quality. But for methods that employ global transformation mapping to construct mosaic, it only needs a few correspondence points to recover the transformation mapping and then a mosaic can be constructed.

The possibility of constructing panorama using the n-image algorithm could be a suitable topic for further research. We cannot construct a panorama using the current system because of the limitation of accumulated image alignment errors. If we want to construct a panorama, the first image and the last image of an image sequence need to be merge together in the process. But for the current system, merging the first image and the last image of an image sequence will cause larger error in the resulting panorama since the position of the last image will be inaccurate due to the accumulated image alignment errors.

Bibliography

[1] E.L. Bras-Mehlman, M. Schmitt, O.D. Faugeras and J.D. Boissonat. *How the Delaunay Triangulation can be used for Representing Stereo Data*. In Proc. of the Second International Conference on Computer Vision, pages 54-63, 1988.

[2] M.T. Cheung and R. Chung. *Mosaic Construction from Image Stream with Parallax*. In Proc. of Second International Workshop on Digital and Computational Video, pages 86-92, February 2001.

[3] F. Dornaika and R. Chung. *Image Mosaicing under Arbitrary Camera Motion*. In Proc. of the Forth Asian Conference on Computer Vision, pages 484-489, January 2000.

[4] O. Faugeras. *Stratification of Three-Dimensional Vision: Projective, Affine, and Metric Rpresentations*. In Journal of Optical Society of America A, pages465-484, Vol12, No. 3, March 1995.

[5] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. The MIT press, 1993.

[6] O. Faugeras. *What can be seen in Tree Dimensions with an Uncalibrated Stereo Rig*. In Proc. of Second European Conference on Computer Vision, pages 563-578, 1992.

[7] S.D. Fleischer, H.H. Wang, S.M. Rock, M.J. Lee. Video Mosaicking along Arbitrary Vehicle Paths. In Proc. of the 1996 Symposium on Autonomous Underwater Vehicle Technology, pages 293-299, 1996.

[8] R. I. Hartley. *Projective Reconstruction and Invariants from Multiple Images*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1036-1041, Vol 16, No. 10, October 1994.

[9] R. Hartley and R. Gupta. *Computing Matched-Epipolar Projections*. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pages 549-555, June 1993.

[10] D. Kalman. A Singularly Valuable Decomposition: The SVD of a Matrix. In The College Mathematics Journal, pages 2-23, Volume 27, No. 1. January 1996.

[11] R. Kumar, P. Ananadon, and K. Hanna. *Direct Recovery of Shape from Multiple Views: a Parallax Based Approach*. In the 12th International Conference on Pattern Recognition, pages685-688, vol. 1, October 1994.

[12] Q.–T. Luong and O.D. Faugeras. *Determining the Fundamental Matrix with Planes: Instability and New Algorithm*. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pages 489-494, June 1993.

[13] R. Mohr, F. Veillon and L. Quan. *Relative 3D Reconstruction Using Multiple Uncalibrated Images*. In Proc. of the IEEE International Conference of Computer Vision and Pattern Recognition, pages 543-548, June 1993.

[14] C. Morimoto and R. Chellappa. *Fast 3D Stabilization and Mosaic Construction*. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pages 660- 665, June 1997.

[15] C. Rothwell, G. Csurka and O. Faugeras. *A Comparison of Projective Reconstruction Methods for Pair of Views*, In Proc. of the Fifth International Conference on Computer Vision, pages 932-937, November 1995.

[16] B. Rousso, S. Peleg, I. Finci and A. Rav-Acha. *Universal Mosaicing using Pipe Projection*. In Proc. of IEEE International Conference on Computer Vision, pages 945-952, January 1998.

[17] J.-R. Sack and J. Urrutia. *Handbook of Computational Geometry*. Elsevier, 2000.

[18] H.S. Sawhney. *3D Geometry from Planar Parallax*. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pages 929-934. June 1994.

[19] H.S. Sawhney, S. Ayer and M. Gorkani. *Model-based 2D&3D Dominant Motion Estimation for Mosaicing and Video Representation*. In Proc. of IEEE International Conference on Computer Vision, pages 583-590, 1995.

[20] H.S. Sawhney, S. Hsu and R. Kumar. *Robust Video Mosaicing through Topology Inference and Local and Global Alignment*. In Proc. of European Conference on Computer Vision, pages 103-119, June 1998.

[21] A. Shashua. *Algebraic functions for recognition*. IEEE Trans. On Pattern Analysis and Machine Intelligence, pages 779-789, Volume 17, No.10, 1995.

[22] A. Shashua. *Projective Structure from Uncalibrated Images: Structure from Motion and Recognition*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 778-790 Vol. 16. No. 8, August 1994.

[23] H. Shum and R. Szeliski. *Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment*. In International Journal of Computer Vision, pages 101-130, Vol 36, No. 2, February 2000.

[24] Z. Zhang. *Determining the Epipolar Geometry and Its Uncertainly: A Review*. International Journal of Computer Vision, pages 43-76, Volume 27, No.2, 1998.

[25] Z. Zhang, R. Deriche, O. Faugeras, and Q.–T. Luong. *A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry*. Artificial Intelligence Journal, pages 87-119, Volume 78, October 1995.

[26] I. Zoghlami, O.Faugeras and R. Deriche. *Using Geometry Corners to Build a 2D Mosaic from a set of Images*. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pages 420-425. June 1997.

[27] www.targetjr.org