MOTION PANORAMA CONSTRUCTION FROM STREAMING VIDEO FOR POWER-CONSTRAINED MOBILE MULTIMEDIA ENVIRONMENTS

by

XUNYU PAN

(Under the Direction of Suchendra M. Bhandarkar)

ABSTRACT

In modern times, more and more multimedia applications are implemented on wireless computer networks and used to entertain users through mobile devices. In power-constrained environments such as pocket PCs, PDAs, and cellular telephones, the large amount of video information transmitted from the server-end to the user-end is often compressed to reduce the power and band width consumption. This thesis introduces an efficient method for the construction of motion panoramas and panoramic videos from streaming video. The technique involves the extraction of motion components from the background mosaic which is generated by a hybrid algorithm that combines both feature-based methods and direct methods. Experimental results show this heuristic approach reduces the size of the video information transmitted and summarizes the entire contents of the motion video for the mobile end users.

INDEX WORDS: Motion Panorama, Panoramic Video, Image Mosaics,
Motion Components Extraction, Multimedia Applications,
Computer Networks, Power-constrained Environments

MOTION PANORAMA CONSTRUCTION FROM STREAMING VIDEO FOR POWER-CONSTRAINED MOBILE MULTIMEDIA ENVIRONMENTS

by

XUNYU PAN

B.S., Nanjing University, China, 2000

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2004

MOTION PANORAMA CONSTRUCTION FROM STREAMING VIDEO FOR POWER-CONSTRAINED MOBILE MULTIMEDIA ENVIRONMENTS

by

XUNYU PAN

Major Professor:	Suchendra M. Bhandarkar
Committee:	Walter D. Potter Khaled Rasheed

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2004

ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisor Dr. Suchendra M. Bhandarkar, for his patient guidance and instructions at every step from the preparation of the research to the draft of this thesis.

I would also like to thank Dr. Walter D. Potter and Dr. Khaled Rasheed for their time and support as members of my committee.

My parents support me all the time. I think I would not complete this without their love and encouragement.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS iv

CHAPTER

1 INTRODUCTION 1

2 OVERVIEW OF THE PROJECT 7

3 STATIC BACKGROUND GENERATION 10

3.1 The Detection of Interest Points 10

3.2 Point-to-point Correspondences 12

3.3 Computation of Homography 14

3.4 Image Blending 21

3.5 Background Mosaic Generation 22

4 FOREGROUND AND BACKGROUND SEGMENTATION 25

4.1 Dynamic Foreground and Static Background 25

4.2 Mahalanobis Distance 26

4.3 Probability Image 27

4.4 False Motion Detection 27

4.5 Segmentation of Dynamic Foreground 29

4.6 Connected Components Detection 31

4.7 Size Filtering 34

4.8 Motion Components Extraction 35

5 MOTION PANORAMA CONSTRUCTION....39

5.1 Network Transmission....39

5.2 Motion Panorama Construction at the User-end....39

6 EXPERIMENTAL RESULTS....42

7 CONCLUSIONS AND FUTURE DIRECTIONS....47

7.1 Concluding Remarks....47

7.2 The Original Contributions of This Thesis....49

7.3 Future Work and Directions....51

BIBLIOGRAPHY....52

CHAPTER 1

INTRODUCTION

Transmission of video streams of large size is always a bottleneck in multimedia applications on computer networks. The requirement of efficient video transmission is a key factor in improving overall system performance, especially in the power-constrained multimedia environments consists of mobile devices such as PDAs, pocket PCs, and cellular telephones.

Automatic construction of large and high quality image mosaics is an active research area in the fields of computer vision, image processing, and artificial intelligence. Efficient methods for mosaic generation can be widely used in networked or mobile applications with the expanded requirement of transmission, storage and manipulation of multimedia information.

The problem of acquiring panoramic images can be solved mainly in two ways, namely:

- Using wide field of view lenses and imaging devices.
- Mosaic construction techniques.

Wide field of view lenses or imaging devices can be used to capture the whole scene of the video sequences, such as Columbia's OmniCam [1, 2]. One shortcoming of this technique is that panoramic images acquired are of low image quality because of the mapping of the entire scene into a fixed resolution video camera. The distortion in the shape of the objects in the scene is another problem introduced by this method.

Mosaic construction technique, which is also called panorama construction, is another approach to display the entire scene. This is an efficient and convenient representation of the motion video by stitching the individual frames into a unique wide-angle panoramic image. It does not require any special imaging devices or hardware. The final panoramic image, which covers the entire scene, does not lose any image quality either. Former works on panoramic mosaics can be divided into two major categories:

- Mosaic construction from static scenes.
- Mosaic construction from dynamic scenes under two situations:

  - Dynamic scenes captured with a static camera.

  - Dynamic Scenes captured with a moving camera.

Static-scene based mosaic construction deals with the situation where the video sequences have static foreground and background. In another words, no obvious motion object is included in the video sequences. A number of papers, e.g. [3], concentrate on this case. Figure 1.1 and figure 1.2 give an example of a panoramic mosaic constructed from a static scene. Figure 1.1 shows several original frames extracted from a video captured in the Visual and Parallel Computing Laboratory (VPCL). The final panoramic mosaic is illustrated in figure 1.2. This panoramic mosaic is generated using a subset of the algorithm implemented in this thesis.

Other researchers have explored efficient methods to represent the dynamic scenes situation where the video sequences of moving objects captured by a static or a moving camera are analyzed. In this case, the scenes are dynamic containing motion or deformation within. Many real-life video sequences are instances of this situation.

The mosaics of dynamic scenes captured by a static camera have been studied for many years and several approaches have been developed [4, 5, 6]. The main idea is to segment the frame into two parts or two layers: foreground and background or dynamic layer and static layer. The moving objects can be extracted by pixel-to-pixel comparison between the pre-stored background and the current frame. These methods work satisfactorily only when the background information is already available which means that the scene should be sampled first.

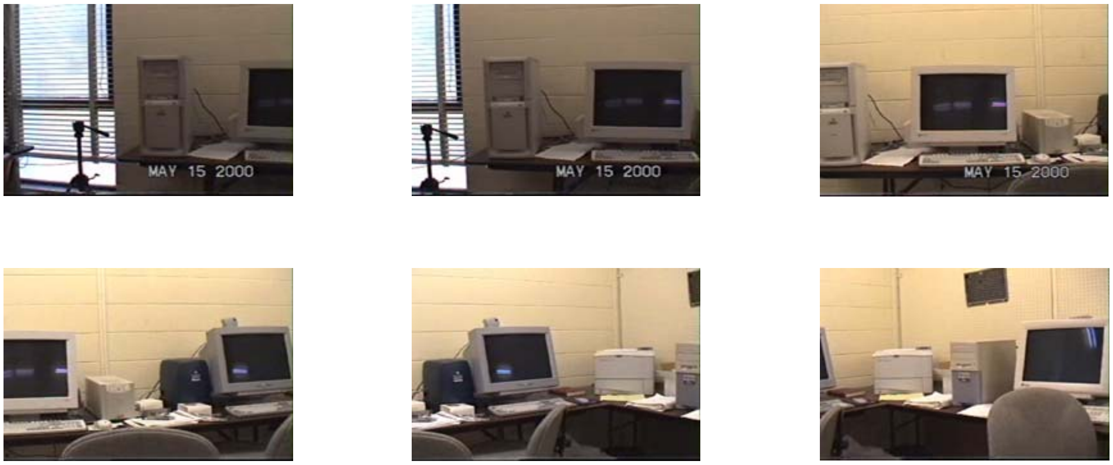


Figure 1.1: The extracted original frames from video sequence

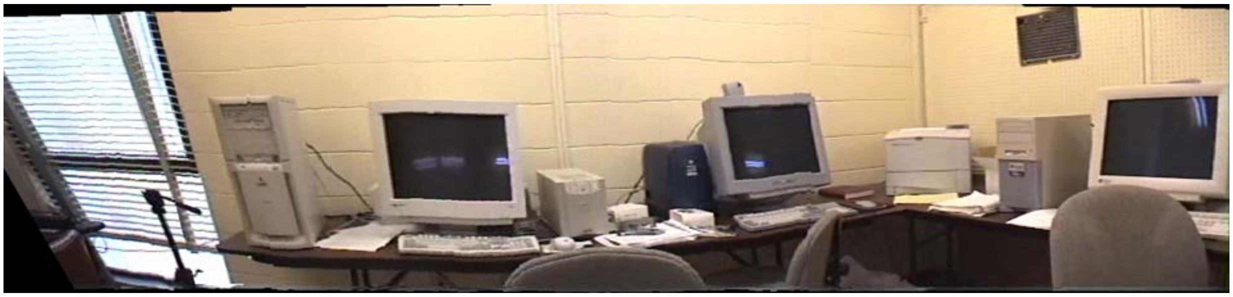


Figure 1.2: The panoramic mosaic constructed from the original frames

To create panoramic mosaics for dynamic scenes captured by a moving camera, [7] [8], and [9] have presented several effective methods. In [9] the authors use the blocking motion detection technique to compute a motion vector field which is then clustered to find the dominant motion regions. In [8], the authors propose a direct method to acquire the motion parameters, align the frames using these parameters, and locate the frame regions which do not observe the motion parameters.

The last case, namely, mosaic construction for dynamic scenes captured by a moving camera, is the most popular situation in real life. In general, two major methods can be applied to this category of problems: feature-based methods [10] and direct methods [11]. The former treats pairs of interest points as features and uses the correspondences of these features in video sequences to estimate the homography between frames. The latter aligns the frame intensity values to acquire the best mapping between frames.

Feature-based methods were introduced by P. H. S. Torr and A. Zisserman in 2000 [10]. These methods involve a strategy for the initial estimation of frame matching which is also called inter-frame homography based on the detection of point features. In other words, the recovery of the entire scene should be achieved by first extracting the features, and then using these features to compute the relations or homographies between the frames. The feature-based methods can also be combined with outlier-rejection techniques such as Random Sample Consensus (RANSAC) [12]. Since the combined methods can estimate the frame matching corresponding to the motion of the camera while rejecting the moving parts of the frame which correspond to a different motion, the technique is quite robust to many real-life situations. However, many alignment problems are caused by the cases where the detected features are not homogeneously distributed across the frames.

Direct methods [11] deal with problems of camera motion and correspondence of every pixel simultaneously. The motion estimation is obtained by this class of methods using measurable information such as brightness variants or image cross-correlation measures. It finds the mapping relations between frames by minimizing the discrepancy between every pixel value in the frame. This category of methods is in contrast to the feature-based methods that rely on the correspondence of a sparse set of highly reliable image features. Since the information of every pixel in the frame is used to estimate frame matching which corresponds to the motion of the camera, the direct methods have better performance in terms of the final mosaic quality.

Both feature-based methods and direct methods contribute to the estimation of motion parameters of the camera between frames. These parameters are essential for the alignment procedure. In particular, feature based methods are more robust in many real-life situations where several motions different from the camera motion, namely outliers, are present in frames. On the other hand, direct methods provide more accurate frame alignment by taking into account every pixel in the frames.

Considering the complementary characters of these two categories of methods, the combined technique that includes both of them is obviously appealing. In this thesis, a combined approach based on both feature-based methods and direct methods is proposed. In practice, a feature-based method is implemented for the static background generation while the direct method is used to segment the dynamic foreground. Some original contributions for amending the drawbacks of these two categories of methods are also provided. The static background generation and dynamic foreground extraction are performed at the server end. During the last phase, the static background and the dynamic foregrounds along with the associated information of their relative locations in the final motion panorama are transmitted to the user-end. At the user-end, the

dynamic foregrounds corresponding to each frame in the video sequence are pasted back onto the static background. Finally, a motion panorama or a panoramic video is constructed under user-specified requirements. Experimental results show that the size of the information transmitted is, on average, from 1/8 to 1/10 of the original motion video. The savings in computation time and memory storage at the user-end are very useful and efficient in power-constrained multimedia environments.

CHAPTER 2

OVERVIEW OF THE PROJECT

The method of motion panorama construction proposed in this thesis is described in three major phases: static background generation, background/foreground (moving objects) segmentation, and motion panorama construction. The first two phases are performed at the server-end. The last phase is performed at the user-end.

The first phase is static background generation. Based on the video sequence extracted from the original motion video, the homographies corresponding to the motion of the cameras are computed for each frame. The static background of the entire scene expressed in the video sequence is generated by stitching the individual frames into a large wide-angle panoramic image using the homographies.

The dynamic foreground which includes regions of both moving objects and false detections existing in the scene is segmented by warping together three consecutive frames in the video sequence and consequently detecting the intensity discrepancy at each pixel. The dynamic foreground is smoothed of noise using a Gaussian filter and then filtered of false motion using a size filter to generate the components of real moving objects. These components are small in size compared to the original frames and hence convenient for network transmission.

After the static background, foreground objects and their location information are received at the user-end, the foreground objects are pasted back to the static background using their location information such as homographies and position coordinates which were computed at the server-end. The final output can be constructed in the form of a motion panorama or a panoramic video determined by the user requirements.

From the perspective of practical application, the whole procedure of motion panorama generation at the server-end, transmission through the mobile network and construction at the user-end is illustrated in figure 2.1.

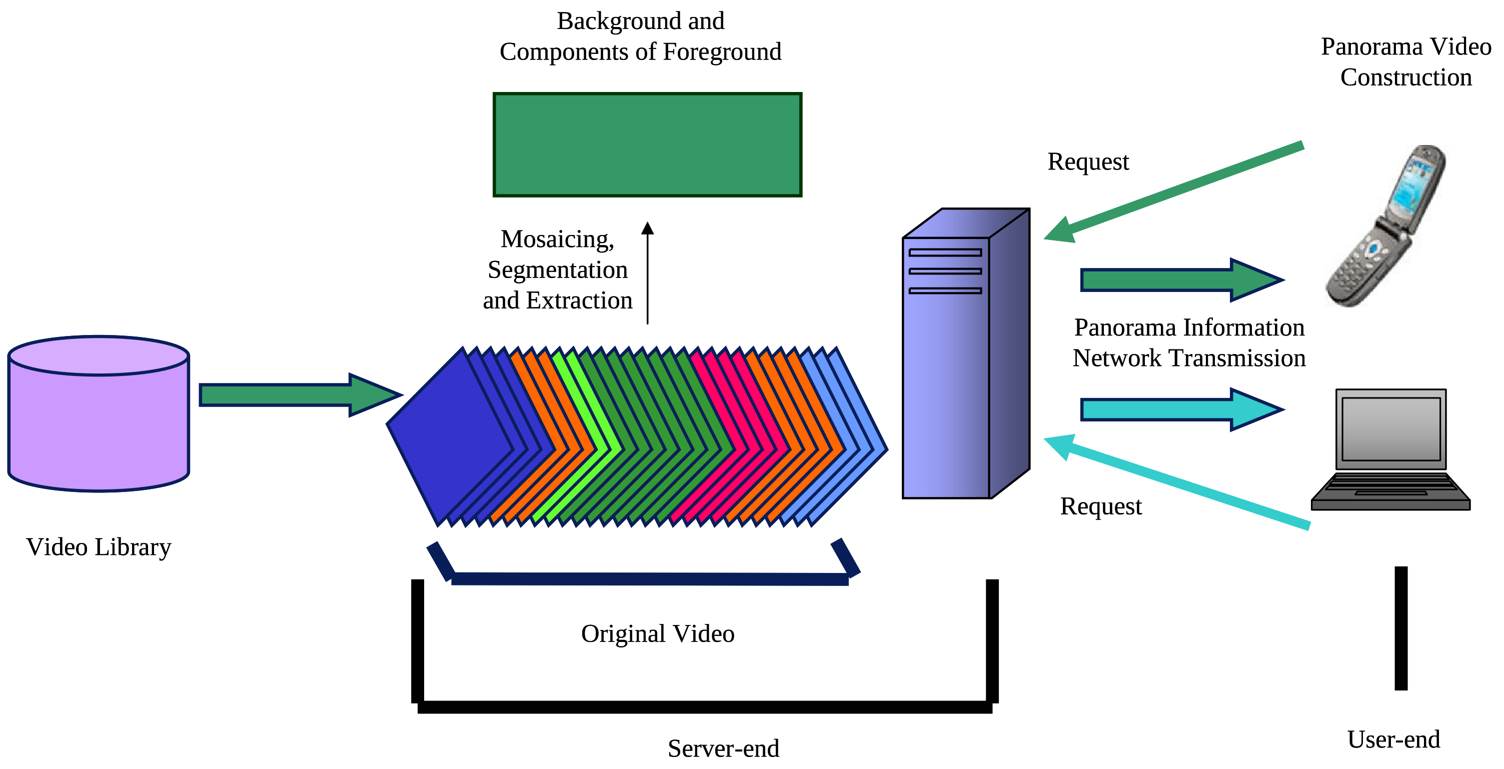


Figure 2.1: The procedure for motion panorama generation, transmission and construction.

CHAPTER 3

STATIC BACKGROUND GENERATION

3.1 The Detection of Interest Points

Knowing the corresponding points between frames enables one to estimate the mathematical expressions for the geometric transformations of frames caused by the motion of the camera such as pan and tilt. The same motion usually holds for most of the pixels in the frames except for these associated with the moving objects. If all possible corresponding points are scanned, the computational complexity usually is very expensive. The process can be simplified by examining only the smaller number of points called interest points. Interest points have some local property. For example, the corners of the objects are good examples of interest points.

Interest points can be detected by a corner detector. Instead of using the Harris corner detector [13] used in feature based registration, the Moravec corner detector [14] is implemented in this thesis. The reasons for using this detector are:

- The detector is effective. Based on the auto-correlation function, it captures the intensity change around a point. A point is detected as an interest point if the change is big enough. This property is helpful to a subsequent cross-correlation matching algorithm which can find the correspondences for the current interest points.
- The detector is simple and computationally inexpensive. The Harris corner detector, another widely used corner detector, calculates the eigen values which usually involves

complex matrix computation. In comparison, the Moravec corner detector is computationally more efficient.

The Moravec corner detector works in the following manner:

(1) The interest value of each pixel in the frame can be calculated by the following equation

$$MO(i,j) = \frac{1}{8}\sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1}\left|f(k,l) - f(i,j)\right| \tag{1}$$

where (i,j) and (k,l) are the coordinates of the pixels in the frame. In the current implementation, 7×7 windows are used to calculate the interest values of every pixel in the frame.

(2) A threshold should be set to filter out the points with relatively small interest value. Only points with large enough interest values can be treated as interest points. In practice, a value of $E+3\sigma$ is used as the threshold, where E and σ are the mean and standard deviation of the interest values of all the pixels in a frame.

(3) To solve the problem of detected interest points that are not homogeneously distributed across the frames, an amended method is adopted. Each frame to be processed is divided into a number of neighbored and non-overlapping 30×30 windows. For each window, the pixel with maximum interest value is extracted as the interest point of this region.

Another issue that needs to be mentioned here is that the interest points extracted by the above method are also called interest features or point features [10]. Since this category of points constitute the basic registration information for frame alignment and the subsequent frame mosaic generation, the so approaches to generate panoramic mosaics based on these point features are called feature-based methods. Figure 3.1 shows the interest points detected by the Moravec corner detector.

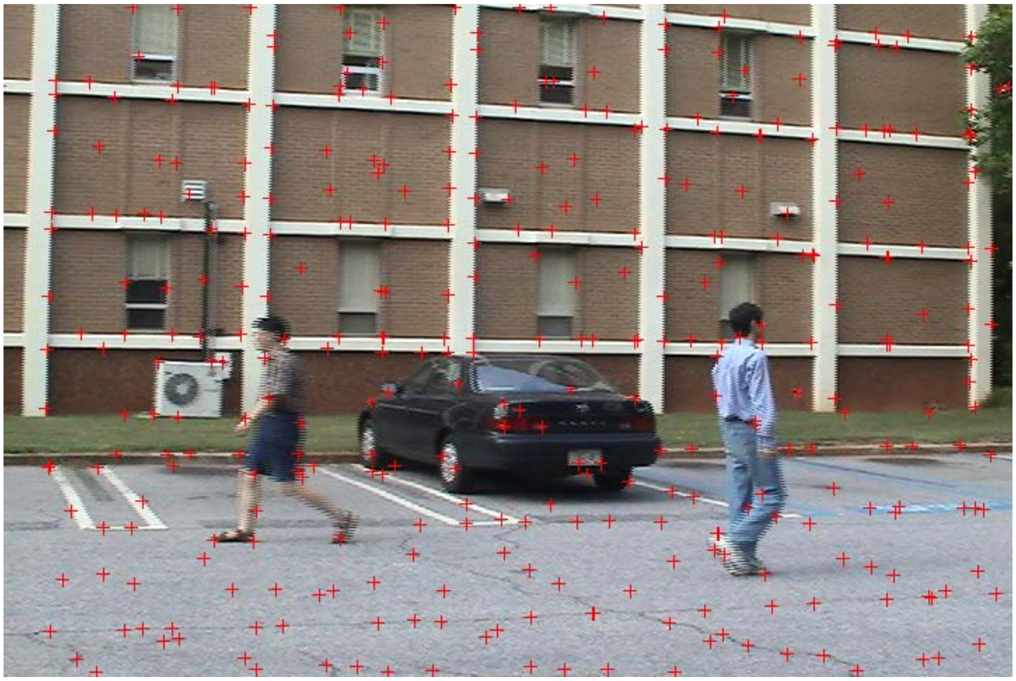


Figure 3.1: Interest points (features) detected by the Moravec corner detector

3.2 Point-to-point Correspondences

The interest points extracted from the frames by the corner detector are then tracked over the video sequence in order to establish the point-to-point correspondences. Template matching [15] is one of widely used method to detect instances of a template in an image frame.

Given a template $t[i, j]$, in order to detect its instances in a frame $f\ [i, j]$, an obvious method is to place the template in the frame and compare the intensity values in the template with the corresponding values in the frame. In many cases, the intensity value will not match exactly. Hence, the sum of squared errors is the most popular matching measure.

Cross-correlation is an operation that can be used to achieve template-matching. Given the interest points extracted from the original frames, the point-to-point correspondences of these interest points are matched using proximity and similarity of the intensity value in their neighborhood. The intensity values of all neighbors of each interest point are used to rank possible matches by computing a normalized cross-correlation. For an $m \times n$ template $t[i, j]$, the match measure M can be computed using

$$C_{ft}[i, j] = \sum_{k=1}^{m} \sum_{l=1}^{n} t[k, l] f[i + k, j + l] \qquad (2)$$

$$M[i, j] = \frac{C_{ft}[i, j]}{\{\sum_{k=1}^{m} \sum_{l=1}^{n} f^2[i + k, j + k]\}^{1/2}} \qquad (3)$$

In the experiment, this method is implemented by using a 15×15 template in the current frame with the interest point at the top left corner. Then a 45×45 region in the next frame is considered as the search area. By moving the template window column by column in the search area, the local maxima where the matching point is the putative correspondence of the interest point under can be found. The point-to-point correspondences between two consecutive frames in the video sequence are illustrated in figure 3.2.

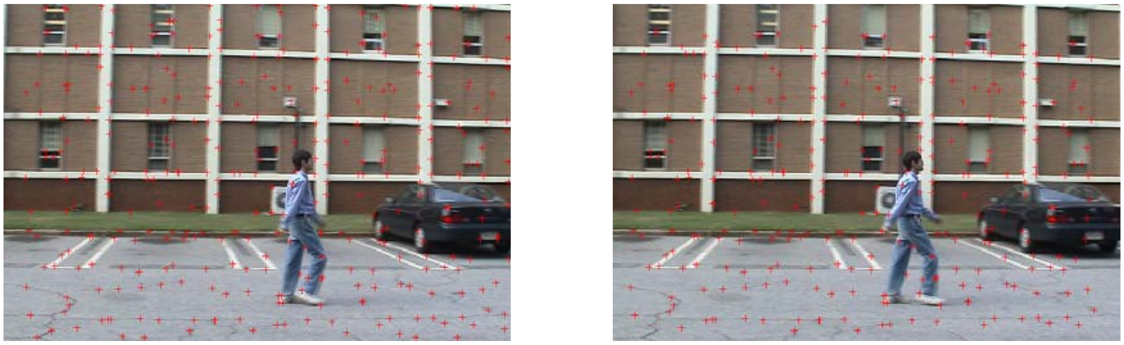


Figure 3.2: The putative point-to-point correspondences between two consecutive frames

The word *putative* indicates that the correspondences detected by the cross-correlation operation are not necessary the real correspondences. It has been reported that more than 40% of the putative correspondences obtained by the best cross-correlation score and proximity are incorrect [10]. Hence robust estimation methods, such as RANSAC which will be described and applied later, are an essential part of the whole procedure of static background generation.

3.3 Computation of Homography

3.3.1 Initial Homography estimation

In real life, people usually use the pin-hole camera to capture the world. This camera model projects the 3-dimensional world onto a 2-dimensional image plane. Let each image to be considered to lie in a projective plane P^2 . Given a set of interest points $\mathbf{x}_i$ in P^2 and a corresponding set of points $\mathbf{x}_i^{'}$ likewise in P^2, the 2-dimensional homography is the projective transformation that maps $\mathbf{x}_i = (x_i, y_i, w_i)^{\mathrm{T}}$ onto $\mathbf{x}_i^{'} = (x_i^{'}, y_i^{'}, w_i^{'})^{\mathrm{T}}$. In practice, $\mathbf{x}_i$ and $\mathbf{x}_i^{'}$ are points in two distinct frames. For a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}$, the problem can be described as being required to compute a 3×3 homography matrix **H** for each *i* such that

$$\mathbf{H}\,\mathbf{x}_i = \mathbf{x}_i^{'} \tag{4}$$

The above equation involves homogeneous vectors and hence the 3-vectors $\mathbf{x}_i^{'}$ and $\mathbf{H}\,\mathbf{x}_i$ are not equal. They have the same direction but may differ in magnitude by a non-zero scale factor. The equation can be expressed by vector cross product as $\mathbf{x}_i^{'} \times \mathbf{H}\,\mathbf{x}_i = \mathbf{0}$**.** If the *j*-th row of matrix **H** is denoted by $\mathbf{h}^{j\mathrm{T}}$, then a simple linear solution for **H** can be derived as follows:

$$\mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} \mathbf{h}^{1\mathrm{T}}\,\mathrm{x}_i \\ \mathbf{h}^{2\mathrm{T}}\,\mathrm{x}_i \\ \mathbf{h}^{3\mathrm{T}}\,\mathrm{x}_i \end{bmatrix}.$$

Suppose $\mathbf{x}_i^{'} = (x_i^{'}, y_i^{'}, w_i^{'})^{\mathrm{T}}$. The cross product may then be given as

$$\mathbf{x}_i^{'} \times \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} y_i^{'}\mathbf{h}^{3\mathrm{T}}\mathbf{x}_i - w_i^{'}\mathbf{h}^{2\mathrm{T}}\mathbf{x}_i \\ w_i^{'}\mathbf{h}^{1\mathrm{T}}\mathbf{x}_i - x_i^{'}\mathbf{h}^{3\mathrm{T}}\mathbf{x}_i \\ x_i^{'}\mathbf{h}^{2\mathrm{T}}\mathbf{x}_i - y_i^{'}\mathbf{h}^{1\mathrm{T}}\mathbf{x}_i \end{bmatrix}.$$

Since $\mathbf{h}^{j\mathrm{T}}\ \mathbf{x}_i = \mathbf{x}_i^{\mathrm{T}}\ \mathbf{h}^j$ for j = 1, 2, 3, this gives a set of three equations in the entries of **H**, which may be written in the form

$$\begin{bmatrix} 0^{\mathrm{T}} & -w_i^{'}\mathrm{x}_i^{\mathrm{T}} & y_i^{'}\mathbf{x}_i^{\mathrm{T}} \\ w_i^{'}\mathrm{x}_i^{\mathrm{T}} & 0^{\mathrm{T}} & -x_i^{'}\mathrm{x}_i^{\mathrm{T}} \\ -y_i^{'}\mathbf{x}_i^{\mathrm{T}} & x_i^{'}\mathrm{x}_i^{\mathrm{T}} & 0^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = 0\,. \tag{5}$$

These equations all have the form $\mathbf{A}_i\,\mathbf{h} = \mathbf{0}$, where $\mathbf{A}_i$ is a 3×9 matrix, and **h** is a 9 vector made up of the entries of matrix **H**.

Although there are three equations in (5), only two of them are linearly independent. In other words, each point-to-point correspondence gives two equations in the entries of **H**. The third equation is usually omitted in computing **H** [16]. Then the set of equations becomes

$$\begin{bmatrix} 0^{\mathrm{T}} & -w_i^{'}\mathrm{x}_i^{\mathrm{T}} & y_i^{'}\mathbf{x}_i^{\mathrm{T}} \\ w_i^{'}\mathrm{x}_i^{\mathrm{T}} & 0^{\mathrm{T}} & -x_i^{'}\mathrm{x}_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = 0\,. \tag{6}$$

The set of equations (6) holds for all points expressed in homogeneous coordinates $\mathbf{x}_i^{'}$ = $(x_i^{'}, y_i^{'}, w_i^{'})^{\mathrm{T}}$, where $w_i^{'}$ = 1, and $(x_i^{'}, y_i^{'})$ are the coordinates of the point in the image.

Given n corresponding points, $2\times n$ such equations can be obtained. A set of four point correspondences yields a set of eight equations which can be written as:

$$\mathbf{Ah} = \mathbf{0},$$

where **A** is the matrix of coefficients built from the matrix rows $\mathbf{A}_i$ contributed by each point-to-point correspondence, and **h** is the vector of unknown entries of **H**. So four point correspondences are the minimum number needed to solve the problem.

In general, given $n \geq 4$ point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}\}$, the homography matrix **H** such that $\mathbf{H}\,\mathbf{x}_i = \mathbf{x}_i^{'}$ can be computed by the Direct Linear Transformation (DLT) Algorithm [17] described by the following steps:

(1) For each correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}\}$ compute the matrix $\mathbf{A}_i$ using equation (6).

(2) Generate a single $2n \times 9$ matrix **A** from the n 2×9 matrices $\mathbf{A}_i$.

(3) Compute the Singular Value Decomposition (**SVD**) of **A** [18]. The unit singular vector corresponding to the smallest singular value is the solution of **h**. In detail, if **A** = $\mathbf{UDV}^{\mathrm{T}}$ with **D** a diagonal matrix with positive diagonal entries, arranged in descending order down the diagonal, then **h** is the last column.

(4) The matrix **H** is determined from **h** as following:

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}.$$

3.3.2 Robust Estimation

For a set of correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}\}$ obtained by the cross-correlation algorithm, the assumption up to now is that the only error is in the measurement of the point's position, which follows a Gaussian distribution. However, in practical situations, two other categories of mismatched correspondences, also called outliers, exist. One category represents the spurious correspondences caused by miscalculation in some special cases. The other consists of the point matches corresponding to moving objects in the scene and not to the motion of the camera. The outliers can severely disturb the estimated homography, and hence should be identified. In real-life applications, robust estimation can deal with the situation where less than 50% of the points in the frame are outliners. Robust estimation is an essential part of homography computation process.

One popular robust estimation technique called the Random Sample Consensus (RANSAC) [12] is used in this thesis. Unlike the classical techniques for parameter estimation such as least-squares that only average the measurement errors, RANSAC has a heuristic mechanism for detecting and rejecting gross errors caused by outliers. For the correspondences detection problem, the faulty measurement of a point's position is a measurement error and follows a Gaussian distribution. This category of errors can be averaged out by classical least-squares techniques. The other two categories of mismatched correspondence, namely spurious correspondence and point matches corresponding to moving objects, are gross errors and can only be filtered out by the RANSAC technique.

The implementation of the RANSAC technique in this project is described by the following steps:

(1) Randomly select 4 correspondences which may include both the correct ones and the mismatched ones to compute a homography **H**. This step constitutes an initial homography computation which has already been described previously in detail in the chapter 3.3.1.

(2) Compute the Euclidean distance for every correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$using the following function:

$$\sqrt{d^2(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)}\ . \tag{7}$$

(3) Compute the number of inliers whose Euclidean distance is less than a threshold *D*. These inliers constitute a consensus set *S*.

(4) If the size of *S* is larger than a threshold *T*, re-compute **H** from the inliers in *S* and terminate.

(5) If the size of *S* is smaller than *T*, repeat the above steps from (1)-(4) for *N* samples.

(6) After *N* samples, recompute **H** from the consensus set with the largest number of inliers.

Several parameters need to be determined here:

- The sampling number *N*. If one chooses to try all possible samples, then $N = C_4^n$, where *n* is number of correspondences. Even for a modest value of *n*, the total number of possibilities could be huge, which implies very expensive computation. Since the try-all-possibilities method is infeasible, *N* can be chosen according to probability *p* which makes that at least one of the random samples of *s* points is not an outlier. Suppose *w* is the probability that any selected data point is an inlier, and hence $\varepsilon = 1 - w$ is the probability that it is an outlier. At least *N* samples can make $(1 - w^s)^N = 1 - p$, so that

$$N = \log(1 - p) / \log(1 - w^s). \tag{8}$$

Because the w and ε are usually unknown, they can be determined adaptively [17] by the following procedure:

```
(1) $N = \infty$, sample_count = 0.
(2) While $N >$ sample_count Repeat
    - Choose a sample and count the number of inliers.
    - Calculate $\varepsilon = 1 - w$.
    - Compute $N$ using equation (8) with $p = 0.99$.
    - Increase sample_count by 1.
(3) Terminate
```

- The Euclidean distance threshold D. Hartley and Zisserman [17] assume that the measurement error is a Gaussian random variable with zero mean and standard deviation σ , and, in this situation, that d^2 is a χ^2 distribution. The probability that a χ^2 random variable is less than any given number k^2 is given by the cumulative chi-squared distribution, F (k^2), which can be found in any standard mathematical table. If k^2 is set to 0.95, $D = 5.99\sigma^2$. In practical experiments, this threshold value is too large and hence not practical. Also, the distribution of measurement error is certainly not Gaussian, since many outliers exist. So a relatively small value of D = 1.25 is chosen, which works well for the experiments.
- The threshold T for the size of an acceptable consensus set. To ensure that the correct model can be found and to satisfy the final smoothing procedure, for n sample points, T = (1 - ε) n is a good choice. For the situation where ε is unknown, a T with value a little larger than that necessary for a smoothing computation can be used.

3.3.3 Optimal Estimation

The homography obtained by robust estimation can be used as a guideline for further optimal estimation. All correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}\}$ between any two frames are calculated by the function given in equation (7). The outliers of these correspondences are filtered out using the same threshold value used in RANSAC procedure.

The correspondences classified as inliers are then used to determine a maximum likelihood estimate of **H** by minimizing the following object function:

$$\sum_i d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}_i^{'})^2 + d(\mathbf{x}_i^{'}, \mathbf{H}\mathbf{x}_i)^2 .$$

In the experiment, a linear least squares method is used to obtain the optimal **H** which best satisfies all the inliers.

3.4 Image Blending

Based on the optimal homography **H**, the frames can be well aligned. However, there are still differences in the intensity values of pixels, which are caused by the changing of the camera's internal parameters during different periods of the capturing process, especially in the regions where the frames overlap. To solve this problem, a function to weight each pixel in all frames is introduced:

$$w(i,\ j) = \left|\frac{h/2 - i}{h/2}\right| \times \left|\frac{w/2 - j}{w/2}\right|,$$

where h and w are the height and the width of the frame. Heuristically, the pixels at the edge of the frames are given less weight.

3.5 Background Mosaic Generation

The homography $\mathbf{H}$ of any two non-consecutive frames is obtained exactly by the composition of homographies of all the frames between them. For example, the homography of the first and the third frame $\mathbf{H}_{13}$ can be computed by the composition of the homography between the first frame and the second frame and the second frame and the third frame as $\mathbf{H}_{13} = \mathbf{H}_{23}\mathbf{H}_{12}$. By using a special frame as the reference frame such as the first frame or the last frame, the homographies between all frames in the video sequence and this reference frame can be computed. Consequently, all frames can be mapped onto the reference frame to generate the background mosaic.

It should be noted that the origin of the generated background mosaic image is different from the frame origin. The origin of the background mosaic shifts during the frame warping procedure. A bounding box for the current mosaic origin should be recorded during the processing of homographies computation. When the computation of homographies between all frames and the reference frame is completed, the lower left corner of the bounding box of the entire mosaic namely $(x_{\min}, y_{\min})$ is obtained. This origin of the entire background mosaic is used to calculate the inverse of the homography matrix used to generate a background mosaic from the frame located at the origin of the entire mosaic. The shifted inverse of any homography is computed by the composition of the inverse of the homography and a translation matrix. For example, $\mathbf{H}_{21} =$ $\mathbf{H}_{12}^{-1}\mathbf{T}$, where $\mathbf{T}$ is given by:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_{\min} \\ 0 & 1 & y_{\min} \\ 0 & 0 & 1 \end{bmatrix}.$$

A sample of the static background panorama generated by the algorithm implemented in this thesis is illustrated in figure 3.3. The main steps of background mosaic generation can be summarized as follows:

(1) Detect the interest points using the Moravec corner detector in each frame.

(2) Find the correspondences between frames using these interest points by a cross-correlation operation.

(3) Compute the initial homographies between frames using the Direct Linear Transformation (DLT) Algorithm.

(4) Use the RANSAC technique to filter out outliers of correspondences for each pair of frames.

(5) Compute a maximum likelihood estimate to obtain the optimal homographies over all frame pairs. Specifically, the linear least squares algorithm is implemented to compute the maximum likelihood estimate based on the inliers of the correspondences.

(6) Use the estimated optimal homographies to stitch all the frames onto a reference frame to generate the static background mosaic.

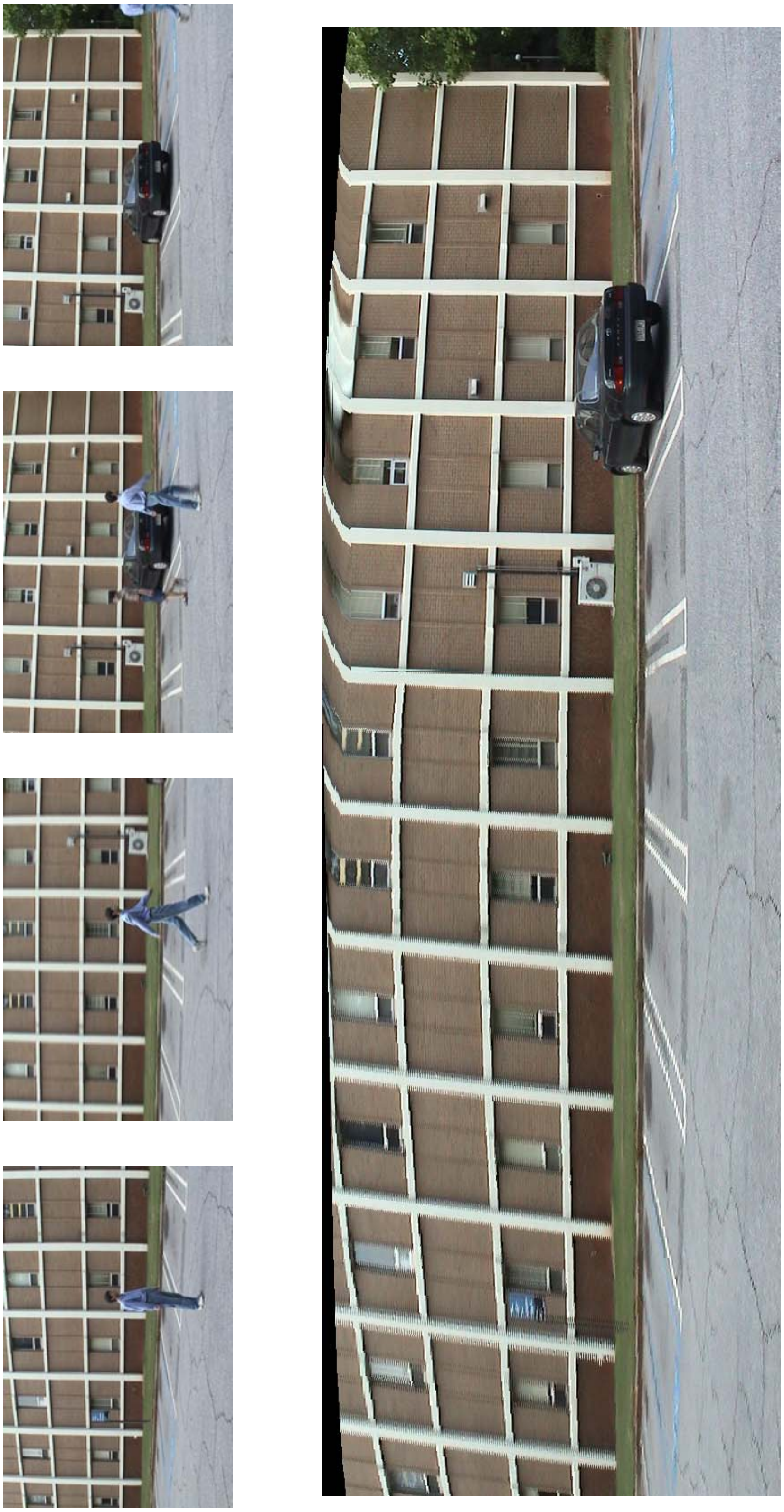


Figure 3.3: The static background generated from the original frames in the video sequence without the dynamic foreground.

CHAPTER 4

FOREGROUND AND BACKGROUND SEGMENTATION

4.1 Dynamic Foreground and Static Background

The frames extracted from the original motion video can be segmented into two layers: static background and dynamic foreground. The static background generated by the procedure described in the previous chapter includes all relatively static objects in the scene such as buildings or mountains. The dynamic foreground that needs to be segmented, on the other hand, is associated with the moving objects such as walking people or moving cars.

Dynamic foreground segmentation is relatively easier for the cases where the dynamic scenes are captured by static cameras. Since the camera is always located in the same position and there is no motion of the camera such as pan and tilt, the moving objects in the scene can be extracted by pixel-to-pixel comparison between the pre-stored background and the current frame being processed. This strategy works well only when the background information is available beforehand.

The foreground segmentation for dynamic scenes captured by moving cameras is computationally much more complex. The camera motions such as pan and tilt usually compensate for the motion of the moving objects in the scene such that these objects remain in the center of the frame. For example, actors or athletes always stay in the center of the images or frames of the movie sequence because the camera is panned or tilted in order to follow them. So a more sophisticated background/foreground separation technique is required to deal with this complex situation.

4.2 Mahalanobis Distance

In the video sequence, the previous and the next frames can be mapped onto the current frame using the estimated homographies. The color values of every pixel in the frame are then compared at each pixel location. The pixels belonging to the static background follow the estimated camera motion and hence the changes of intensity value between the corresponding pixels are relatively small. On the other hand, large discrepancy in intensity values occurs at pixels which do not conform to the estimated homography. The comparison of color values at each pixel location is achieved by the following distance function:

$$d(\mathit{\Gamma}_{i\text{-}1}(q_{i\text{-}1}),\mathit{\Gamma}_i(q_i))+d(\mathit{\Gamma}_{i+1}(q_{i+1}),\mathit{\Gamma}_i(q_i)) \tag{9}$$

where $\mathit{\Gamma}_i(q_i)$ is the intensity value of the pixel q in the frame $\mathit{\Gamma}_i$, and i is the number of frames in the video sequence. Note that d is the Mahalanobis distance [19] which represents the discrepancy of in color values between the two pixels when they appear in two consecutive frames. The Mahalanobis distance is given by

$$d(\mathit{\Gamma}_{i\text{-}1}(q_{i\text{-}1}),\mathit{\Gamma}_i(q_i))=(\mathit{\Gamma}_{i\text{-}1}(q_{i\text{-}1}),\mathit{\Gamma}_i(q_i))^{\mathrm{T}}\mathbf{C}^{-1}(\mathit{\Gamma}_{i\text{-}1}(q_{i\text{-}1}),\mathit{\Gamma}_i(q_i))$$

where **C** is the covariance matrix for the RGB color space, and is estimated using red, green, and blue color values for all the pixels and for each frame in the video sequence.

4.3 Probability Image

Based on the values obtained for each pixel location in the frame computed using the function in equation (9), a probability image is generated. It is made up of the likelihood of every pixel in the frame belonging to the dynamic foreground. For example, a large discrepancy in color value at the same pixel position q in three consecutive frames has a large probability of being the dynamic foreground. Here the three consecutive frames are any set of previous frame, current frame and next frame in the video sequence. The pixels belonging to moving objects which do not conform to the homography between consecutive frames have a greater probability of possessing larger discrepancy in color values.

4.4 False Motion Detection

Although the above approach, which borrows ideas from the direct method to mosaic reconstruction, can detect the dynamic layer of the frame, there are still problems with it. Two categories of false detection of moving objects exist in the video frames. One is caused by a certain level of pixel-level noise which is introduced by the camera capture or the video production process. The other category is caused by the presence of large homogeneous regions and complex motions such as articulated body motions which are widely present in many real-life videos. Large homogeneous regions are the interior of the moving objects. The articulated body motions are characterized by the fact that some body parts move while some other body parts remain still.

The problem of the presence of false motion can be solved by performing a Gaussian filtering on the probability image. The Gaussian smoothing filter is very well suited for removing noise that is drawn from a normal distribution. In the context of image processing, the two-dimensional zero-mean discrete Gaussian filter is given by

$$g[i,j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

and is used as a smoothing filter. A typical two-dimensional Gaussian filter is illustrated in figure 4.1.

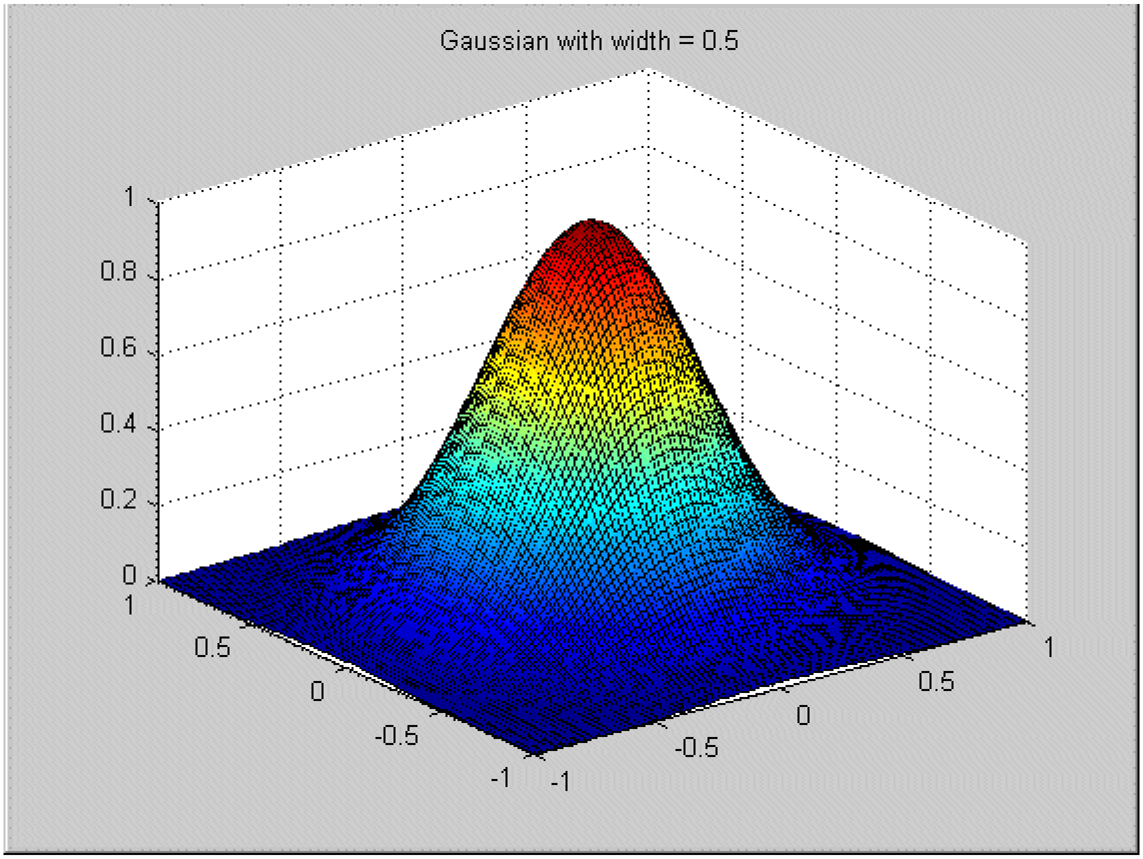


Figure 4.1: A typical two-dimensional Gaussian filter

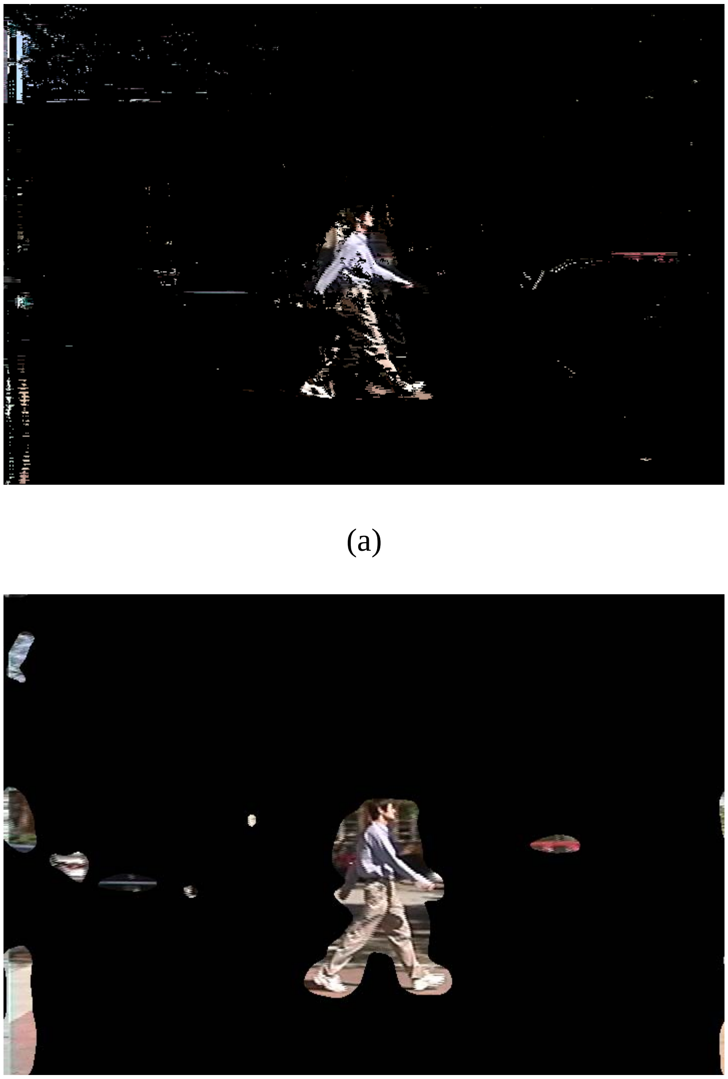
The smoothing procedure is performed on the probability image instead of the original frame. Although Gaussian smoothing is essentially a procedure to blur the image, applying the filter on the probability image can retain the image quality as well as filter out the noise. The detected regions corresponding to the moving objects prior to smoothing and the regions corresponding to the moving objects after smoothing by a Gaussian filter are illustrated in figure 4.2.

4.5 Segmentation of Dynamic Foreground

To detect the motion foreground from the original frames, a probability threshold is set to optimistically segment the dynamic layer from the static layer. The threshold is applied onto the probability image which has been smoothed by the Gaussian filter.

In general, a probability value of less than half of the maximum in the probability image is suggested as the threshold. In the experiment, the recommended threshold is so large that many parts of motion objects were deleted. So in practice, a value of 1/8 of the maximum value is used. When applying this threshold to the probability images, the pixels with the probability value larger than the threshold were classified as the dynamic foreground and kept. At the same time, the pixels which are smaller than the threshold were classified as the static background and removed.

Though the dynamic foregrounds have been segmented, they are still stored with the background in the image file. The only difference is that all the pixels belonging to the background change from the original color to black. In order to reduce the multimedia information transmitted through the mobile network, the detected dynamic foreground within each frame could be divided further into several connected components and then extracted from the background.



(a)

(b)

Figure 4.2: (a) The frame showing the detected regions corresponding to the moving objects prior to smoothing; (b) The detected regions corresponding to the moving objects after smoothing by a two-dimensional Gaussian filter.

4.6 Connected Components Detection

All the frames in the video sequence are now divided into two layers: the static background layer and the dynamic foreground layer. To find all the connected components in the dynamic foreground which includes both real moving objects and the noisy or spurious regions, each frame is converted to a binary image where the value “1” is assigned to the pixels of the dynamic foreground and the value “0” is assigned to the pixels of the static background. Here a connected component is a set of pixels in which each pixel is connected to all other pixels in that set.

Unlike the gray scale image, binary image contains only two gray levels. The advantages of binary image include they are well understood and tend to be less expensive and faster during the procedure of image processing than the gray level or color images. Binary images are used in binary vision systems to reduce the memory and computing power requirement. Traditionally, pixels of assumed objects which could include both moving objects and static objects are set to white while the other pixels belonging to background are set to black.

In this thesis, the binary image such as figure 4.3 was generated by set the color of pixels belonging to static background to black and the color of pixels belonging to dynamic foreground to white.

The iterative connected-component labeling algorithm is applied to the binary image and usually requires two passes over the image. This algorithm checks the two neighbors of a current pixel, namely, the one above and to the left of the current pixel and tries to assign an already used label to the current pixel. When the two neighbors have different labels, an *equivalence table* is used to keep track of all labels that are deemed equivalent. This table is used in the second pass to assign a unique label to all the pixels of a connect component.

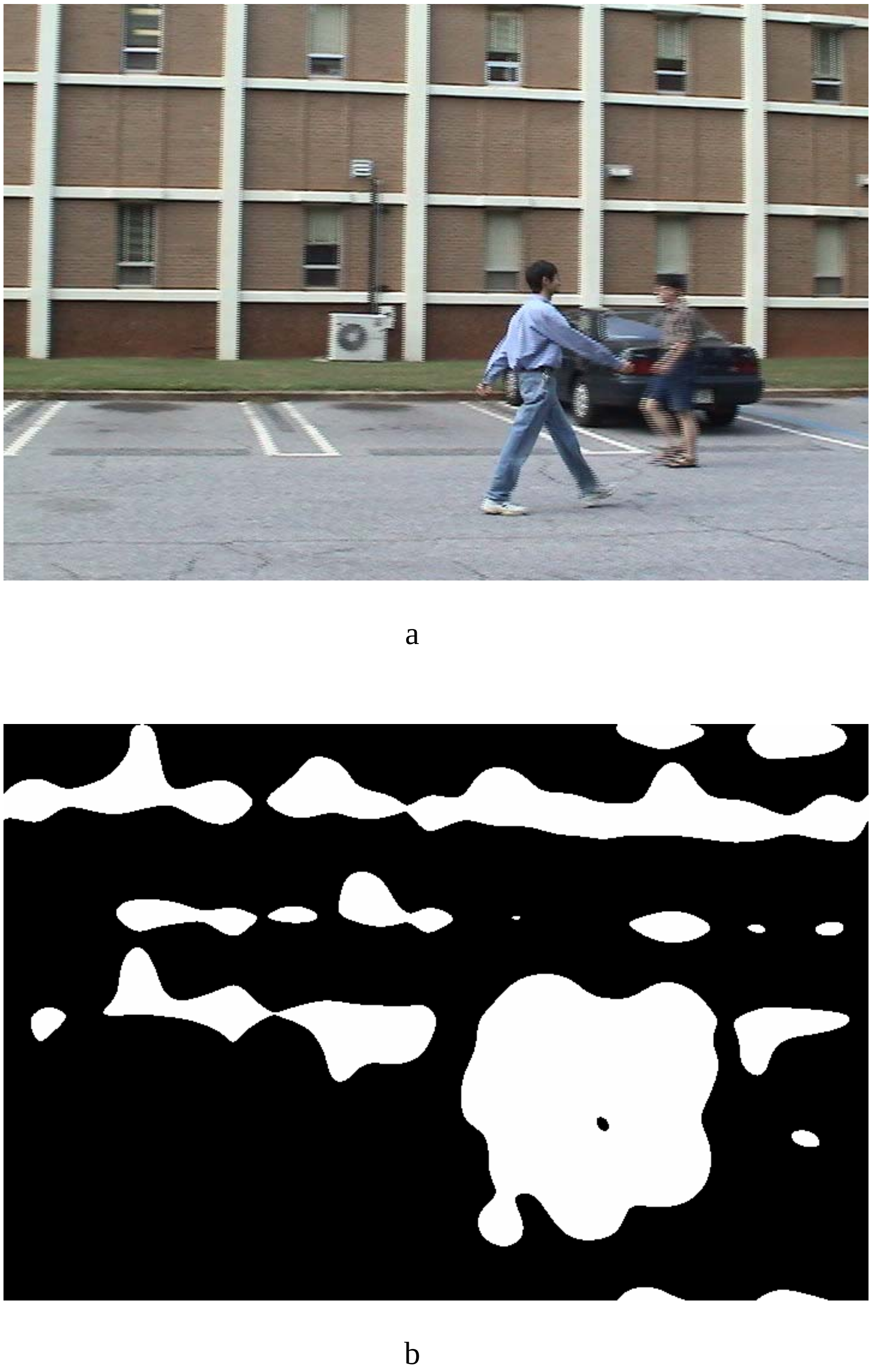


Figure 4.3: (a) The original frame; (b) The corresponding binary image which has already been smoothed by a Gaussian filter.

The algorithm divides the neighborhood relation of pixels into three cases and assigns different labels for them. The equivalence table includes the information of unique labels for each connected component. During the first scan, all labels assigned to one component are claimed as equivalent. In the second pass, the smallest corresponding label from the equivalence table is selected to be assigned to all pixels of a certain component.

When all connected components have been detected, the equivalence table is renumbered to eliminate the gaps between labels. The connected components in the image are then reassigned the new label under the direction of the equivalence table.

The main steps of the iterative connected-component labeling algorithm are summarized as follows:

(1) Scan the binary image from left to right, top to bottom.

(2) If the current pixel is “1”, then

  (a) If only one of its upper and left neighbors has a label then copy that label.

  (b) If both of them have the same label, then copy the label.

  (c) If both of them have different labels, then copy the label of upper pixel and note in a *equivalence table* that label (upper) = label (left).

  (d) Otherwise assign a new label to the pixel and note the label in the *equivalent table*.

(3) Repeat steps (2) (a) - (2) (d) until all “1”-pixel have been visited.

(4) For each equivalence class in the *equivalence table*, assign a unique label, typically the lowest.

(5) Rescan the image and replace the label of each “1”-pixel by the label of its equivalence class.

The above algorithm detects all the connected components in an image. Many properties of the component such as size, position and bounding box can then be computed for each component for later processing.

4.7 Size Filtering

Even after the segmented dynamic foreground has been smoothed by the Gaussian filtering, a certain number of noisy or spurious regions still persist. The motion components are found by the Mahalanobis distance method which detects the motion based on the color discrepancies of the corresponding pixels in the consequent frames. Sometimes the small changes in reflectance and illumination characteristics of other objects in the scene can lead to incorrect detections of the motion. One important property of these spurious regions is that their sizes are small compared with those of the real moving objects in the scene and hence can be removed by a size filter.

The connected components detected by the iterative connected-component labeling algorithm consist of components belonging to both the real moving objects in the scene and the unexpected noises. In order to remove these noises, a size filter is used based on the size property of this category of noise outlined above. When all connected components have been found in the dynamic foreground, the size filter is used to suppress the noisy artifacts with relatively small size in terms of number of pixels.

The threshold of the size filter can not be set to large, which will remove the real moving components. At the same time, the threshold can not be set too small, which will keep too many noises. Considering the different applications, the algorithm should be robust to different cases.

In the experiments presented in this thesis, a threshold of 1/3 or 1/4 of the maximum size of the component in the dynamic foreground is used. The result is illustrated in figure 4.4.

4.8 Motion Components Extraction

After the size filtering operation, only large components corresponding to moving objects are kept. A bounding box which is composed of the minimum and maximum coordinates of a certain component in the frame is recorded into an information file for later transmission. The thresholded components are extracted from the original frames and used to generate a set of small image files which store only the pixels corresponding to regions in the bounding boxes. The procedure for the extraction of these small images corresponding to moving components is shown in figure 4.5.

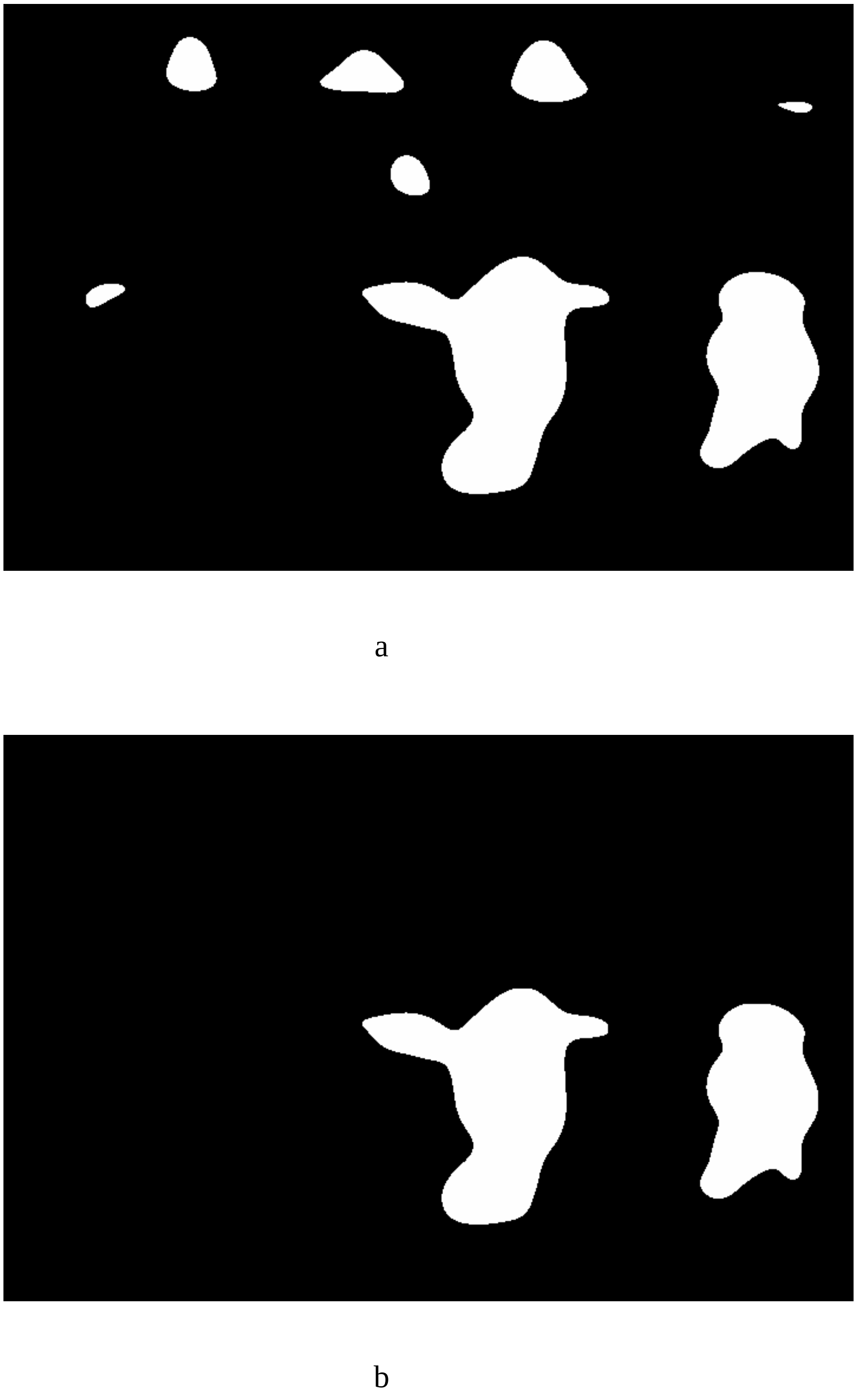
Based on the experimental observations, the small image files representing the moving components in the frame are only, on average, 1/4 to 1/5 of the original frame in terms of size. The compression rate is satisfactory. This implies a very good compression ratio for multimedia information and is really convenient for mobile networked transmission.

The main steps of the foreground and background segmentation procedure which is performed at the server-end can be summarized as follows:

(1) Compute the Mahalanobis distance in color space for every pixel in all the frames using any three consecutive frames.

(2) Generate the probability image for each frame in the video sequence based on the Mahalanobis distance and equation (9).

(3) Use the Gaussian filter to smooth out regions of false motion caused by large homogeneous regions and complex motions.



a

b

Figure 4.4: (a) The original binary image; (b) The binary image filtered by the size filter.

(4) Set a probability threshold to classify each pixel in the frame as belonging to the dynamic foreground or the static background.

(5) Use iterative connected-component labeling algorithm to detect connected components in the binary image of the segmented dynamic foreground.

(6) Apply the size filter to remove the noisy artifacts and identify components belonging to the real moving objects in the video stream by the application of Mahalanobis distance.

(7) Generate certain numbers of sets of small image files corresponding to motion components of real moving objects in each frame in the video sequence. Find the bounding box of the detected motion components and extract the related location information for the later mobile networked transmission.

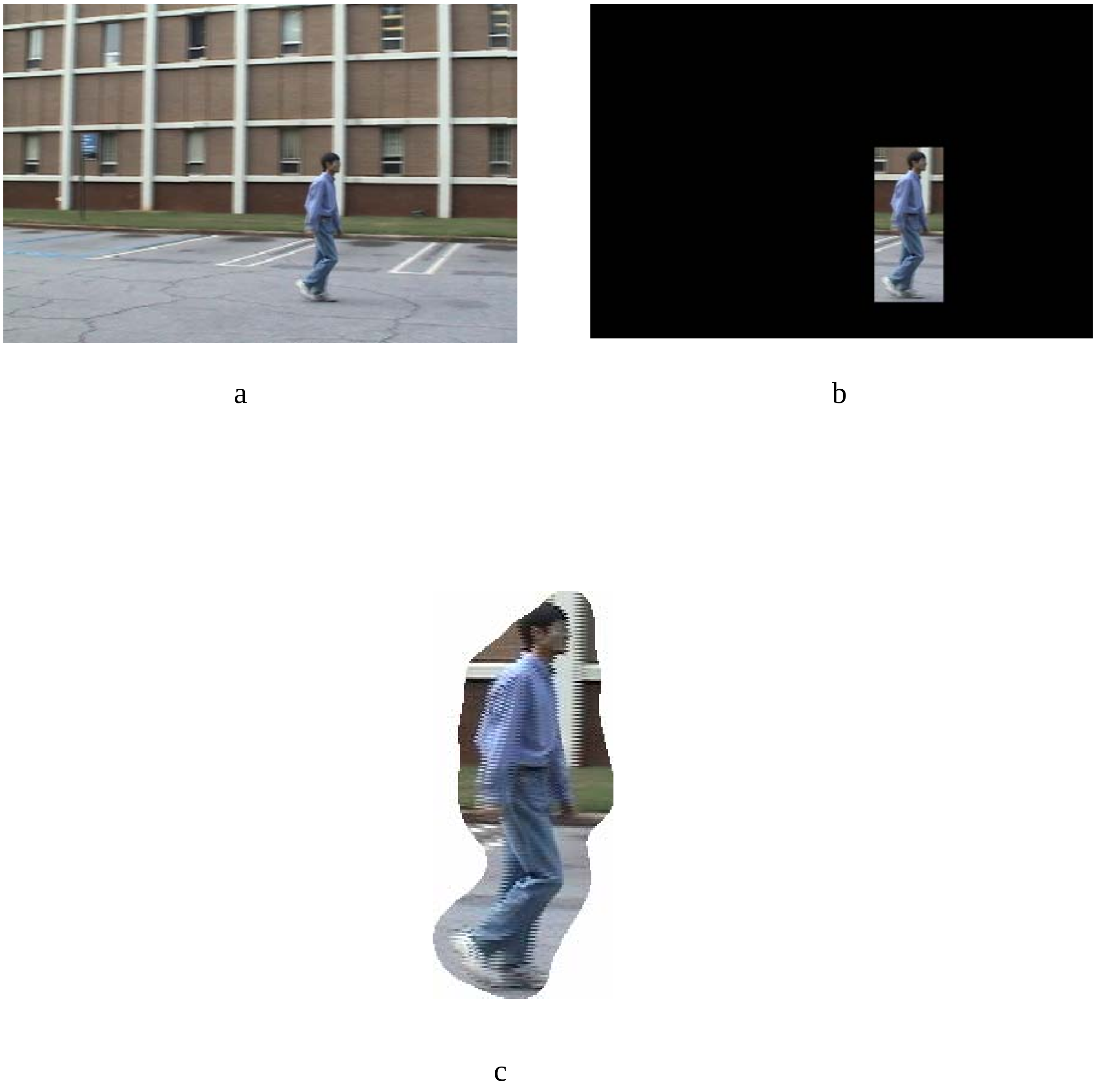


Figure 4.5: (a) The original frame; (b) The bounding box for a moving component; (c) The extracted small image of the region belonging to the moving object.

CHAPTER 5

MOTION PANORAMA CONSTRUCTION

5.1 Network Transmission

Three categories of files are transmitted through the network, namely:

- A single large image file containing the static background. The file includes all background information in the scene captured by the moving camera.
- A certain number of small image files containing the dynamic foreground. These files include all the various components corresponding to moving objects in the scene for each frame in the video sequence.
- An information file for each frame. The file includes all associated parameters such as bounding boxes of dynamic components and the homography between each frame in the video stream and the reference frame.

5.2 Motion Panorama Reconstruction at the User-end

When all the information has been transmitted from the server to the user-end, it is used to reconstruct a motion panorama. The static background image and the dynamic foreground for each frame in the video sequence are available now. The dynamic foregrounds are then pasted onto the static background based on the parameters in the information file to reconstruct the motion panorama.

The homographies between each frame in the video sequence and a reference frame are computed during the procedure of static background generation. The dynamic foreground of each individual frame is mapped onto the background mosaic using these homographies. This is almost the same procedure as the previous generation of background mosaic except that only the extracted regions of foreground are now pasted instead of the entire original frames.

In [10], the authors propose a method to build the background panorama by considering each potential pixel in the background image plane. For each of these pixel locations, the contributions from a certain number of frames are accumulated and weighted to obtain the final intensity value for that pixel. The individual frames are then mapped onto the background and consequently used to extract the dynamic foreground. This method entails a significant amount of computation because all pixels in the large background image, which includes the pixels from both the static background and the dynamic foreground, are determined via the computation of an average of the corresponding pixels from 20 related frames in the video sequence.

In this thesis, the regions comprising the dynamic foreground in each individual frame are segmented from the background. Subsequently, only these foreground regions are pasted onto the static background to reconstruct the motion panorama. More specifically, the segmented components of the dynamic foreground in each frame are mapped onto the background using both, the bounding boxes which include the location information of the dynamic components in each frame, and the homography between that frame and the reference frame. In the static background image, when the intensities of pixels in the mapped regions of dynamic components are the same as those of corresponding pixels in the background, the intensity of the corresponding pixel in the background image does not change. Otherwise, the intensity of a pixel in the background image is replaced by the intensity of the corresponding pixel in the mapped regions representing the dynamic components. In other words, the regions corresponding to the dynamic foreground or moving objects are pasted onto the static background image.

Following the video sequence, if the dynamic foreground is pasted onto the static background once in every few frames, a motion panorama is generated. A static representation of this form containing a large background image with a series of motion objects in it expresses the content of original motion video with much less space. For the application where the panoramic video is required, an alternative strategy is implemented. The dynamic foreground of each individual frame in the video sequence is pasted onto the background separately. Each frame in the video sequence generates one motion panorama. When the generation of panorama images from all frames is completed, one can combine all these images of panorama together to create an MPEG or AVI format file for viewing.

CHAPTER 6

EXPERIMENTAL RESULTS

The technique for motion panorama construction described in this thesis is applied to several motion videos captured by a digital camcorder. The scenes of these motion videos were acquired on the campus of the University of Georgia. A typical sample used in the experiment is a 10 second video with multiple persons walking in front of Dawson Hall. The video, which includes around 210 frames, is 41.25 M bytes in size. The results shown in figure 6.1 and figure 6.2 consist of the procedure of the motion panorama construction using the proposed approach. The panorama is constructed with both the large static background and dynamic foregrounds extracted for every 40 frames.

The panoramic video which consists of a single static background and a certain number of foregrounds corresponding to each frame of the original motion video can also be constructed at the user-end using the similar technique. In this form of representation, the dynamic foregrounds move in a single large background without losing any information from the original motion video.

In table 6.1, three forms of motion representation, namely, original motion video, motion panorama and panoramic video are compared in terms of the type and the size of files based on the multimedia information transmitted through the mobile computer network. The results are satisfactory with an average compression rate of around 0.1. The technique of motion panorama or panoramic video construction can greatly reduce the amount of information transmitted and hence conserve the power consumed at the user-end in power-constrained mobile networked environments.

Table 6.1: The comparison of three forms of motion representation based on the type and the size of files transmitted through computer networks

	Original Motion Video	Motion Panorama	Panoramic Video
Transmitted File(s) with Type and Size	1 file of the motion video-AVI (41.25 M Bytes) or 210 files of original frame-JPG (17.85 M Bytes, average 85 K Bytes per file)	1 file of the static background mosaic-JPG (165 K Bytes)	1 file of the static background mosaic-JPG (165 K Bytes)
		5 set of files of the dynamic foreground-JPG (49.9 K Bytes, average 9.98 K Bytes per file)	210 set of files of dynamic foreground-JPG (3.92 M Bytes, average 18.7 K Bytes per file)
		5 files of associated location information-TXT (0.42 K Bytes, average 84 Bytes per file)	210 files of associated location information-TXT (20.6 K Bytes, average 98 Bytes per file)
Total Size	41.25 M Bytes / 17.85 M Bytes	215.32 K Bytes	4.11 M Bytes

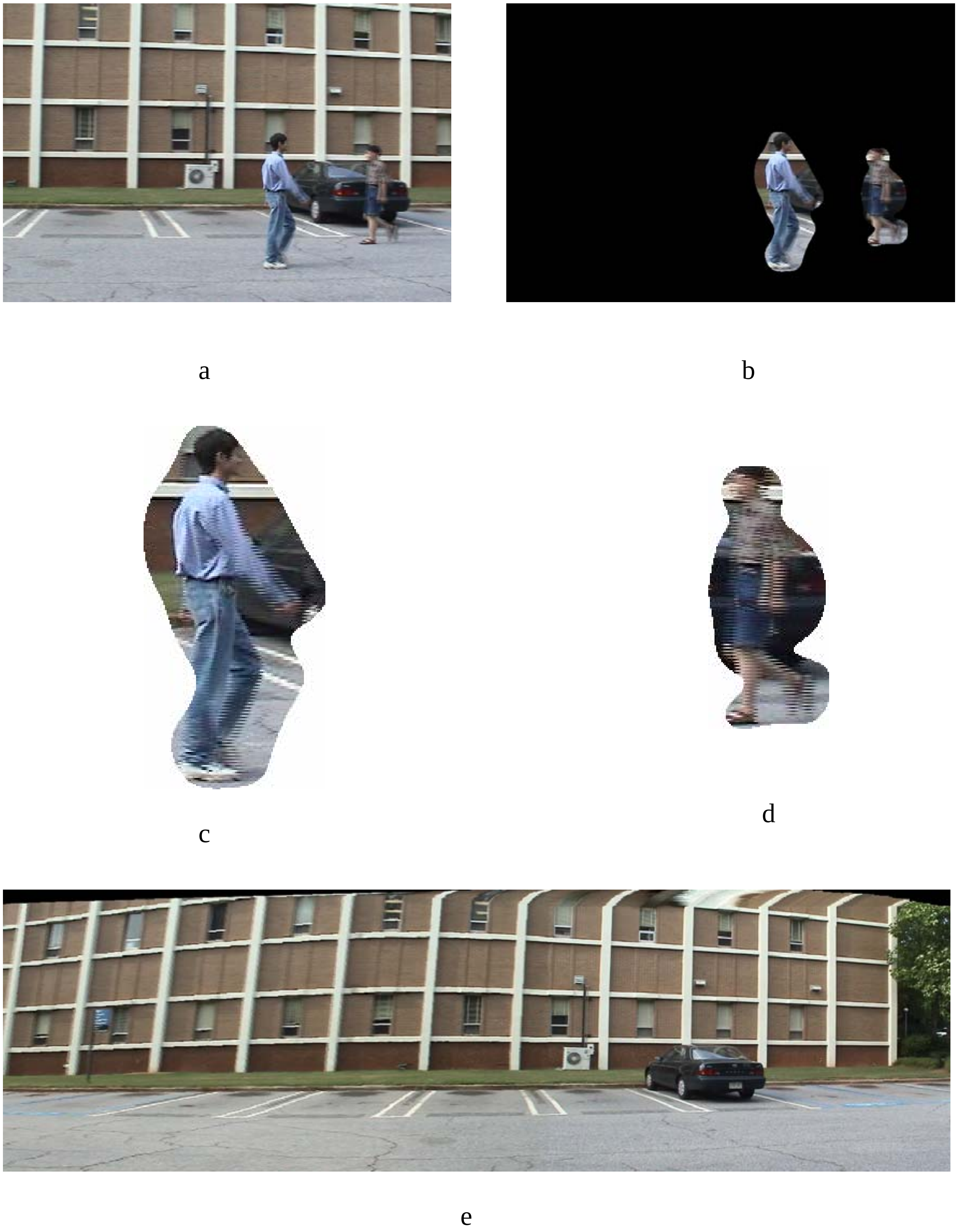


Figure 6.1: (a) An original frame (138); (b) The detected moving components and their location information; (c) and (d) Extracted small images of moving component; (e) The single image of static background. (c), (d) and (e) are the actual files transmitted from the server-end to the user-end.

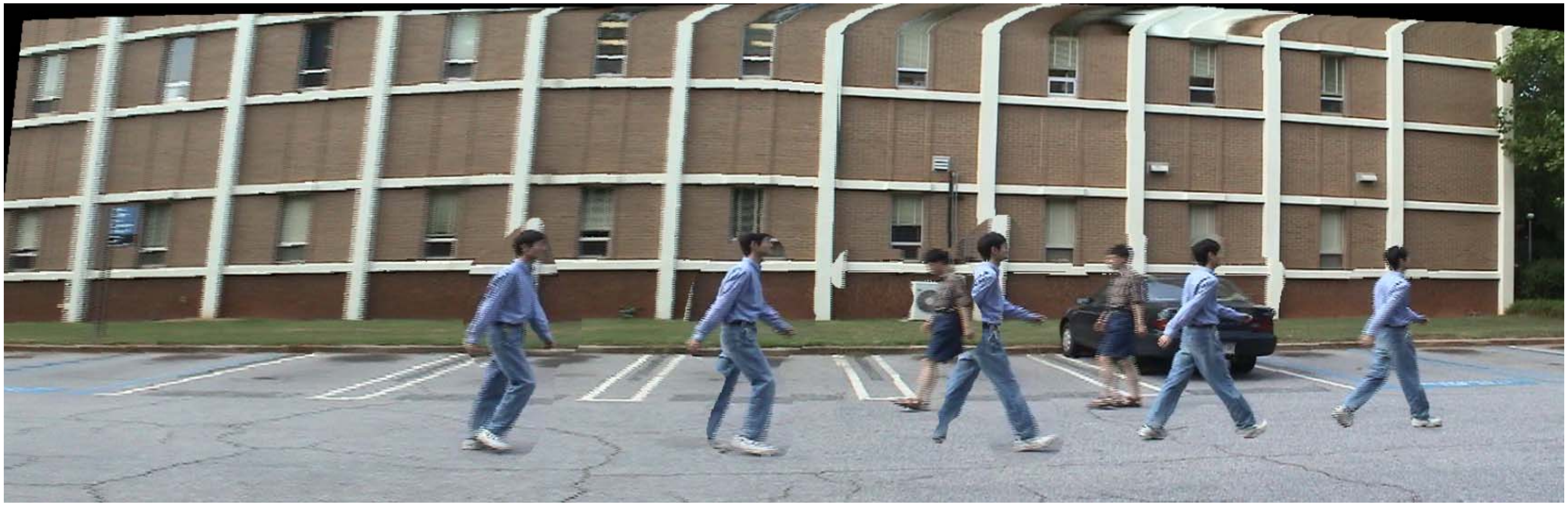


Figure 6.2: The motion panorama with multiple moving objects, which is generated based on the method proposed in this thesis. This panorama is constructed using one large static background and a certain number of dynamic foreground objects extracted once in every 40 frames in the motion video sequence which was captured in front of the Dawson Hall on the University of Georgia campus.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Concluding Remarks

The motion panorama or motion mosaic is a compact and convenient representation for videos of a dynamic scene. In the preceding chapters, a combined method for motion panorama construction and its application in power-constrained environments is described.

Many research articles have reported great success in generation of static image mosaics. Feature-based methods and direct methods are two categories of static image mosaic generation techniques which have been widely accepted and used. However, the mosaic of motion video and its application are novel research issues in the fields of computer vision and artificial intelligence. In this thesis, a modified approach combining the advantages of both feature-based methods and direct methods is proposed to construct a motion panorama from the original motion video. A practical moving-components-extraction technique is also implemented with excellent information compression results compared to the size of the original motion video. In this procedure, the feature-based method and the direct method are applied in different phases at the server-end to segment the dynamic foreground and static background while the motion panorama is constructed by combining the information from both the background and the foreground at the user-end. Some updated and improved algorithms are also implemented during the experiments.

The method for motion panorama construction in power-constrained mobile networked environments described in this thesis involves three major processing phases.

The technique starts with the generation of a static background image at the server-end. The features or interest points are first detected in individual frames using the Moravec corner detector. Based on the correspondences of features identified by the cross-correlation operation, the estimated homographies between frames are computed by using the DLT algorithm with the feature outliers filtered out using the RANSAC procedure. The maximum likelihood estimate of these homographies is then computed using the linear least squares algorithm. Using these optimal homographies, the static background is generated by warping all frames onto a reference frame. In the background generation phase, the feature-based method is applied with several improvements. The Moravec corner detector is used instead of the Harris detector for the purpose of reducing computational complexity. Unlike traditional methods which consider only an abstract interest value, the detection of interest points is achieved using a certain number of small windows which are distributed uniformly across the frame. This property is very important for homography estimation, because it takes the possible motion from all pixel locations into consideration.

The second phase consisting of foreground and background segmentation is also implemented at server-end. First, the Mahalanobis distances are computed for every pixel in each frame using three consecutive frames. Every frame in the video sequence then generates a probability image based on the sum of the Mahalanobis distances between the previous frame and current frame, and the current frame and next frame. A Gaussian smoothing filter of certain width is applied on the probability images in order to filter out the regions of false motion. Each smoothed probability image is classified as the dynamic layer or the static layer using a threshold value that is selected to be less than half of the maximum probability value in each frame. The connected components corresponding to both real moving objects and noisy artifacts are extracted from the binary image of the segmented dynamic layer. A size filter is then applied on the connected components to remove the noisy artifacts which have not been smoothed out by the Gaussian filter. Finally, only the components belonging to the real significant moving objects are extracted.

Given the static background and segmented components of moving objects and their associated location information transmitted from the server-end, the final phase of panorama generation is implemented at the user-end by pasting components belonging to the dynamic foreground onto the static background under the guidance of the associated location information. For different application requirements, the final output could be a motion panorama or a panoramic video.

7.2 The Original Contributions of This Thesis

This thesis applies the techniques for motion panorama construction from streaming video to the mobile networked transmission in power-constrained environments. The following original contributions have been made in this project.

- Based on static image mosaic generation techniques such as feature-based methods and direct methods, a combined approach for motion panorama construction is introduced. This technique performs the static background generation using the feature-based method and the dynamic foreground segmentation using the direct method.
- Some improved algorithms are implemented compared with the original technique introduced by feature-based methods and direct methods. The Moravec corner detector is

applied to detect feature points instead of the Harris detector and hence reduces the computation complexity. A certain number of small neighbored and non-overlapping windows which are uniformly distributed across the image are used to extract the features from all pixel locations. This algorithm solves the problem that the features detected by feature-based methods are not homogeneously distributed in the images which may cause alignment problems.

- For the purpose of reducing the amount of information transmitted through the mobile network, the iterative connected-component labeling algorithm is performed to detect the connected moving components within the image. A size filter is used to remove the noisy artifacts which have not been smoothed out by the Gaussian filter and mistakenly detected as the moving objects. This method can significantly eliminate the number of noises which are not corresponding to components of the real moving objects.
- The single static background image and segmented components of moving objects and their associated location information are transmitted from the server-end to the user-end. An algorithm is proposed to construct the motion panorama using the above information at the user-end. An alternative panoramic video can also be constructed under user-specified requirements.
- The amount of information transmitted through the mobile network is, on average, from 1/8 to 1/10 of the original motion video. This compression ratio for multimedia information greatly reduces the time and space requirements at the user-end. In other words, the techniques used in this thesis conserve the computation time and memory storage at the user-end in power-constrained multimedia environments.

7.3 Future Work and Directions

Although the compression rates of the information for network transmission are satisfactory, the motion panorama can still be improved in terms of image quality. The major characteristics of usually moving objects such as people, animals and mobiles can be stored beforehand and checked during the procedure for dynamic foreground extraction. Using this heuristic method, the moving objects can be differentiated from noisy artifacts more easily and precisely.

In general, since the internal parameters of the camera such as the focal length and aspect ratio are not required to be known, the technique for motion panorama construction is general and flexible and can be used in a wide range of real-life applications. The possible application areas include:

- Robots download the panorama from control headquarters and study the motion information. They can simulate various human actions and execute them in several scenarios such as industrial production, family service and crime detection/prevention.
- Image mosaic and motion mosaic can be used to reconstruct dynamic scenes under various illumination conditions for virtual reality applications.
- Motion panorama techniques can also be applied to the reconstruction of the planet surface during space exploration and in biomedical imaging where a detailed anatomical atlas can be generated by mosaicing a series of snapshots with limited field of view.

BIBLIOGRAPHY

[1] Shree. K. Nayar, Catadioptric Omnidirectional Camera, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '1997),* San Juan, Puerto Rico, Page 482-488, June 17-19, 1997.

[2] Shree K. Nayar and Amruta Karmarkar, 360×360 Mosaics, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '2000)*, Hilton Head, SC, USA, Volume 2, Page 2388, June 13-15, 2000.

[3] H.-Y. Shum and R. Szeliski, Systems and experiment paper: Construction of Panoramic Image Mosaics with Global and Local Alignment, *International Journal of Computer Vision*, 36(2):101–130, February 2000.

[4] M. Irani, P. Anandan, and S. Hsu, Mosaic Based Representations of Video Sequences and Their Applications, In *Proceedings of the Fifth International Conference on Computer Vision (ICCV 95)*, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, Page 605-611, June 20-23, 1995.

[5] H.S. Sawhney, S. Ayer, and M. Gorkani, Model-based 2D&3D Dominant Motion Estimation for Mosaicing and Video Representation, In *Proceedings of the Fifth*

International Conference on Computer Vision (ICCV '95), Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, Page 583-590, June 1995.

[6] J.Y.A. Wang, E.H. Adelson, and U. Desai, Applying Mid-Level Vision Techniques for Video Data Compression and Manipulation, In *Proceedings of SPIE on Digital Video Compression on Personal Computers: Algorithms and Technologies*, San Jose, California, Volume 2187, Page 116-127, February 1994.

[7] J.M. Odobez and P. Bouthemy, Robust Multiresolution Estimation of Parametric Motion Models, *Journal of Visual Communication and Image Representation*, Volume 6, (4):348-365, December 1995.

[8] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, Efficient Representations of Video Sequences and Their Applications, *Signal Processing: Image Communication*, Volume 8:327-351, 1996.

[9] F. Dufaux and F. Moscheni, Background Mosaicking for low bit rate video coding, In *Proceedings IEEE International Conference on Image Processing (ICIP '96)*, Lausanne, Switzerland, Volume 1, Page 673-676, September 16-19, 1996.

[10] P.H.S. Torr and A. Zisserman, Feature Based Methods for Structure and Motion, Estimation, *Vision Algorithms: Theory and practice*, Springer-Verlag, 2000.

[11] M. Irani and P. Anandan, About Direct Methods, In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, Corfu, Greece, Pages 267-277, September 21-22 1999.

[12] Martin A, Fischler and Robert C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the ACM*, 24(6):381-395, June 1981.

[13] C. Harris and M. Stephens, A Combined Corner and Edge Detector, In *Proceedings of the 4th Alvey Vision Conference*, University of Manchester, Pages147-151, 1988.

[14] H.P. Moravec, Towards Automatic Visual Obstacle Avoidance, In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI '77)*, Cambridge, MA, USA, Page 584, 1977.

[15] Ramesh Jain, Rangacher Kasturi and Brian Schunck, *Machine Vision*, MIT Press and McGraw-Hill, March 1995.

[16] I.E. Sutherland, SketchPad: A Man-machine Graphical Communication System, In *Proceedings of the Spring Joint Computer Conference*, Detroit, Michigan, USA, Pages 323–328, May 1963.

[17] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, June 2000.

[18] William H. Press, Saul A. Teukolsky, William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[19] D.C. Alexander and B.F. Buxton, Statistical Modeling of Colour Data, *International Journal of Computer Vision* 44 (2):87-109, September 2001.

[20] A. Bartoli, N. Dalal and R. Horaud, Motion Panoramas, *Research Report 4771*, THE French National Institute for Research in Computer Science and Control (INRIA), Grenoble, France, March 2003.

[21] Bo Hu, Christopher Brown and Andrew Choi, Acquiring An Environment Map through Image Mosaicking, *Technical Report 786*, The University of Rochester, November 2001.