USC-SIPI REPORT #377

Low Complexity Mosaicking and Up-Sampling Techniques for High Resolution Video Display

by

Ming-Sui Lee

December 2006

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

Dedication

To my beloved family

Acknowledgements

It is with grateful thanks to my advisor C.-C. Jay Kuo for his guidance and encouragement during the whole work of this dissertation. I would also like to thank Akio Yoneyama san and Tomoyuki Shimizu san from KDDI Laboratories, Inc., Japan, for their precious comments and collaboration. I'm very grateful to Meiyin Shen, Yu Hu and May Kuo for their kindly help. Also, I'd like to give special thanks to Chia-Hao Chiang for his fully support and encouragement throughout the whole process.

Table of Contents

List of Tables

List of Figures

Abstract

Several challenging issues for applications of image/video mosaicking and upsampling with high resolution are addressed here, all of which are mainly conducted in DCT (Discrete Cosine Transform) domain so that lower computation complexity can be achieved.

First of all, color matching and compensation techniques are proposed to remove the seam lines between image boundaries due to the different color tones of the inputs. Color deviation of each input image is corrected first and color differences between input images are then compensated using the polynomial-based contrast stretching technique. The proposed approach is attractive for its lower computational complexity. Experimental results demonstrate that the color-matching problem can be satisfactorily solved in the compressed domain even when the DCT blocks of original input images are not aligned.

Two block-level image registration techniques for compressed video such as motion JPEG or the I-picture of MPEG are investigated. The proposed methods are based on edge estimation and extraction in DCT domain so that the computational cost of image registration is reduced dramatically as compared with the pixel-domain edge-based registration techniques while achieving certain quality of composition. In order to reach higher accuracy of registration, a post-processing technique, hybrid block/pixel level alignment, is

proposed so that the displacement vector resolution can be enhanced from the block level to the pixel level. As compared with the traditional spatial-domain processing, we do not perform the inverse DCT transform to the whole image but to some selected blocks. It is shown by experiments that the proposed algorithm saves around 40% of the computational complexity while achieving the same quality.

In the last part, a content adaptive technique is proposed to upsample an image to an output image of higher resolution. The proposed technique is also a block-based processing algorithm that offers the flexibility in choosing the most suitable up-sampling method for a particular block type. Block classification is first conducted in the DCT domain to categorize each image block into several types: smooth areas, textures, edges and others. For the plain background and smooth surfaces, simple patches are used to enlarge the image size without degrading the resultant visual quality. The unsharp masking method is applied to the textured region to preserve high frequency components. Since human eyes are more sensitive to edges, we adopt a more sophisticated technique to process edge blocks. That is, they are approximated by a facet model so that the image data at subpixel positions can be generated accordingly. A post-processing technique such as 1D directional unsharp masking can be used to enhance edge sharpness furthermore. Experimental results are given to demonstrate the efficiency of the proposed techniques.

Chapter 1

Introduction

1.1 Significance of the Research

Since the first camera was invented in 1816, there have been great interests in developing more advanced image/video capturing devices and technologies. Many innovative ideas have been brought forth: from analog to digital signals, from monochromatic to color systems, from images to video clips, and from low resolution to high resolution video. Digital video has become popular recently as evidenced by the quick growth of the consumer electronic markets, including DVD (digital versatile disk), DTV (digital television) and other related entertainment products and services.

DTV is a new broadcasting technology that supports multiple digital television formats. Among all formats, high definition television (HDTV) offers the highest quality. HDTV uses a wide screen format and supports very high resolution that contains more than twice as many lines as current analog TVs. Although the HDTV display device becomes affordable to the household these days, HD video capturing devices are prevalently used by professionals due to the high cost of the equipment. There is a growing demand for

general consumers to generate their own high quality multimedia content at a low price. One way to create high resolution video is through video authoring using the image/video mosaic technique.

Image/video mosaicking is the process of stitching two or more images/videos taken by different cameras from different viewpoints. Applications of image/video mosaic techniques can be found in computer vision, pattern recognition and remotely sensed data processing. When input image/video contents are taken from different viewpoints, sampling times and sensors, image registration is needed to integrate these image/video tiles together. Over the past few decades, a lot of research has been done to obtain an image/video mosaic. For an extensive survey of previous work, we refer to [7], [56]. Generally speaking, the image registration technique consists of two major steps: feature detection and feature matching. They will be reviewed in Chapter 2.

Even though image/video mosaicking has been studied for years, most techniques were primarily developed using the information of raw video (or called uncoded video). They are implemented in the space-time domain (or the image pixel domain). In this research, we consider mosaicking of coded video since each individual captured video content is often coded before its transmission. If we perform image/video mosaicking in the raw video domain, we have to perform image/video decoding first. This involves inverse DCT when the input video is in the compressed format such as motion JPEG and MPEG. The approach may not be suitable for the implementation in real-time embedded systems due to a much larger memory requirement and the extra decoding procedure demanded.

For nowadays applications, multimedia capturing and display devices of different resolutions can be easily connected by networks and there is a great need in developing

techniques that facilitate flexible image/video format conversion and content adaptation among heterogeneous terminals. Quality degradation due to down-sampling, up-sampling, blurring, coding/decoding and some content adaptation mechanism in the transmission process is inevitable. Thus, techniques for super resolution and image enhancement are required to generate high quality multimedia outputs. Super resolution and image enhancement both have been investigated for a long time and are able to acquire contents with high performance. The challenge under the current context is to strike a balance between low computational complexity and high quality of resultant image/video.

The above observation has motivated us to study image/video mosaicking and super resolution enhancement directly in the compressed domain. Since motion JPEG, MPEG and H26x coding standards all adopt the DCT representation in the coding process, our goal of this research is to conduct the registration process in the DCT (Discrete Cosine Transform) domain to generate the corresponding high-quality compressed image/video mosaic from multiple compressed video inputs.

1.2 Comparison of Raw and Coded Image/Video Mosaic Techniques

Most traditional image/video mosaic techniques are conducted in the raw image/video domain, which is essentially an image registration problem. Image registration usually consists of two steps: feature extraction (or detection) and feature matching. They are briefly discussed below.

Feature detection can be done either manually or automatically. Since human eyes are sensitive to geometric patterns, it is straightforward for people to choose matched patterns. However, it is desirable to develop an automatic feature selection process based on the particular application context. Feature detection techniques can be classified into two categories: the feature point-based and the area-based approaches. The feature point-based approach extracts salient points such as corners, line intersections, line ends and centroids of closed-boundary regions. For example, the wavelet transform was used in [19] to extract the local maxima. The partial derivatives of image pixel values were proposed in [25] for corner detection. However, this process is time consuming and sensitive to noise. The area-based approach uses the correlation function to determine the degree of closeness of two regions. For example, it computes the cross-correlation of intensities of regions of input images to find the best match. This approach is suitable for images that do not have many details. However, its computational complexity is still high. Once the feature information is available, the next step is to find the optimal correspondence between extracted features. Feature matching is a process to determine the relationship between similar objects contained by different images. This can be achieved by finding the spatial relations between extracted features of input images.

Although existing methods lead to good results under a high SNR (Signal-to-Noise-Ratio) environment, they are only applicable in the raw image/video domain, where the operations are applied to image pixels. This process is computationally expensive in general. In practice, it is seldom to have raw video contents in storage and/or transmission in real world applications. Once some image/video content is captured, it is compressed (or coded) for storage and/or transmission. Commonly used video coding standards such

as motion JPEG, MPEG and H26x all adopt the DCT representation in the coding process. Thus, it is desirable to conduct the registration process directly in the DCT domain for multiple coded video inputs to synthesize an image/video mosaic, which is the main difference between this research and the traditional image/video mosaic techniques.

1.3 Comparison of Image-based and Block-based Super Resolution Techniques

Similar concerns to the raw image/video mosaic techniques, image-based super resolution techniques suffer intensive computation complexity although high performance is guaranteed. Block-based processing is preferred due to several advantages. First of all, block-based algorithm reduce the degree of freedom dramatically. The dimension of the block that is taken into account at a time is much smaller compared to the original image size. Also, the block-based method provides a mechanism which is capable of segmenting the image into several types so that content adaptive image processing can be chosen accordingly. Moreover, since each block can be treated as a smaller image individually, parallel processing is applicable to speed up the processing time even more. Flexibility and low computation complexity of the block-based algorithm make it more attractive than the traditional image-based algorithms.

1.4 Contributions of the Research

In this proposal, we first consider the color matching problem of two input image/video content. Then, we study the image registration of two arbitrarily translated images/videos in the DCT domain. Specific contributions of this research are highlighted below.

- Development of DCT-domain color adjustment techniques

  Several color matching algorithms that compensate color differences from two input image sequences captured by different cameras is proposed. Some of them are conducted in the pixel domain while others are carried out in the DCT domain. The two proposed techniques, *i.e.* histogram matching and polynomial contrast stretching, can eliminate the seam lines successfully at a much low computational complexity as compared with the color adjustment technique in the pixel domain. Moreover, when comparing two proposed approaches, the polynomial-based contrast stretching method outperforms the histogram matching method in terms of the processing time and the memory requirement since solving a second order linear system is faster than performing histogram adjustment and only three matching coefficients are needed to be stored (rather than the whole histogram matching table).

- Development of DCT-domain registration techniques

  The DCT-domain registration techniques for MPEG video are developed for video mosaic authoring with indoor and outdoor scenes. Both of them can achieve certain quality of composition while the computational cost can be reduced significantly in comparison with the pixel-domain based techniques. Furthermore, a post-processing technique called "hybrid block/pixel level alignment", which is conducted partially

in the pixel domain, is introduced to enhance the accuracy of the alignment. For hybrid block/pixel level alignment, an algorithm is proposed to detect corner blocks based on the DCT coefficients. Only corner blocks detected in the DCT domain are converted back to the spatial domain for alignment fine-tuning. This hybrid technique can achieve excellent alignment results at the cost of slightly increased complexity.

- Robustness of DCT-domain registration in a low SNR environment

  It is observed that the proposed DCT-domain registration techniques are robust in the presence of noise. This phenomenon is studied and explained. Rather than dealing with pixel intensities directly, the proposed DCT-domain methods adopt the DC component of the DCT coefficients for block alignment. The DC coefficient can be viewed as the down-sized version of the original image since it is the average energy of the whole 8×8 DCT block. The DCT-domain algorithms are robust in a low SNR environment since noise can be removed by the averaging process.

- Development of DCT-domain block classification techniques

  Properties of 8×8 DCT blocks are investigated for the purpose of block classification. A decision tree is formed to decide whether a block contains background, texture or edges. The decision is made by some thresholds defined based on the energy distribution of DCT coefficients. By following the decision tree, an image can be classified into several categories, which helps reduce the computational complexity of further processing such as super resolution. For example, a simple interpolation scheme can be used in the background region and smooth surfaces. Texture synthesis

techniques can be performed in the texture area. A more sophisticated algorithm with high performance should be adopted for blocks that contain important visual information, such as corners and edges since human eyes are more sensitive to these features.

- Development of DCT-domain super resolution and image enhancement techniques

  A content adaptive technique is proposed to upsample an image to an output image of higher resolution in this work. The proposed technique is a block-based processing algorithm that offers the flexibility in choosing the most suitable up-sampling method for a particular block type. Block classification is first conducted in the DCT domain to categorized each image block into several types: smooth areas, textures, edges and others. For the plain background and smooth surfaces, simple patches are used to enlarge the image size without degrading the resultant visual quality. The unsharp masking method is applied to the textured region to preserve high frequency components. Since human eyes are more sensitive to edges, we adopt a more sophisticated technique to process edge blocks. That is, they are approximated by a facet model so that the image data at subpixel positions can be generated accordingly. A post-processing technique such as 1D directional unsharp masking can be used to enhance edge sharpness furthermore.

1.5 Outline of the Dissertation

This dissertation is organized as follows. The problem of raw and coded image/video mosaicking is explained in Chapter 2. Several DCT-domain algorithms of color matching

and adjustment are presented in Chapter 3. DCT-based image registration techniques are proposed in Chapter 4. Properties of DCT-based image/video registration techniques are investigated in Chapter 5. A content-adaptive up-sampling technique for image resolution enhancement is proposed in Chapter 6. Finally, concluding remarks and future research directions are given in Chapter 7.

Chapter 2

Research Background: Raw and Coded Video Mosaicking

Image/video mosaic, which combines several image/video inputs into a panorama output, has been widely used in image processing, computer graphics, computer vision, and remotely sensed data processing. For a generic scenario, we may consider multiple video sources captured by an arbitrary number of cameras with different parameter settings. The discrepancies among smaller video tiles have to be resolved for seamless composition.

2.1 Problems in Image/Video Mosaicking

Due to different camera calibrations, special attention has to be paid on compensating those disparities such as temporal synchronization, focal length reparation, image registration and color difference adjustment. These issues are described in detail below.

- **Temporal Synchronization**

Consider two input image sequences used to form a video mosaic. The first problem encountered is that temporal sampling points of these two sequences are different. As shown in Fig. 2.1, there is a gap between sampled frames in these two sequences. The goal of temporal synchronization is to perform temporal alignment between the two sequences, which can be achieved by camera calibration or temporal frame interpolation so that the time difference between sequences is significantly reduced.

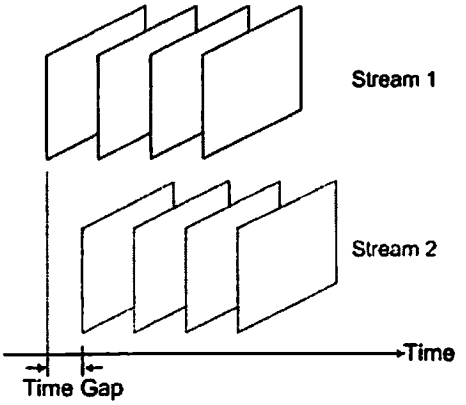
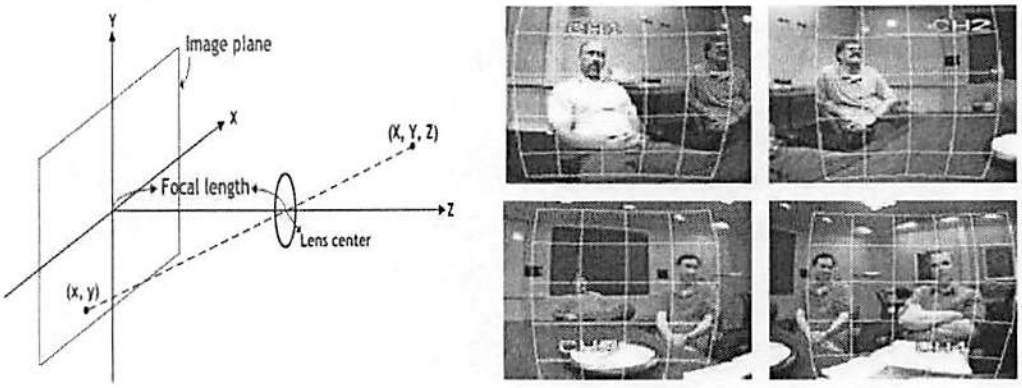


Figure 2.1: Temporal synchronization required for video mosaicking.

- **Focal Length Compensation**

The focal length is the distance along the optical axis from the lens center to its focus (or focal point) as shown in Fig. 2.2(a). The longer the focal length, the smaller the field of view and the smaller the radial distortion. Radial distortion is a lens aberration in which the focal length varies radially outward from the center. It makes a straight line curved around the border of an image, which is also called barrel distortion. An example of various distortion effects of different cameras is

shown in Fig. 2.2(b) [14]. Since the distortion would affect the quality of the mosaic output, it has to be corrected as well.



(a) (b)

Figure 2.2: Need of focal length compensation: (a) illustration of the focal length and (b) barrel distortion effects.

- **Image/Video Registration**

  Image registration is a technique that aligns several partly overlapped images properly so as to create a panorama view. An example is shown in Fig. 2.3. In this step, the critical features of each image have to be detected and their correspondence have to be found to determine the disparities of all images. The disparities between images may include translation, rotation and scaling effects. Translation means that there exists a displacement vector along vertical, horizontal or both directions between a pair of two images. By rotation, we refer to an angle difference between the axis systems of two capturing systems. The scaling effect, also known as the zoom-in and zoom-out effects, is a result of the focal length change. Once the disparities are determined, input images can be aligned so as to form an image with a larger filed of view.

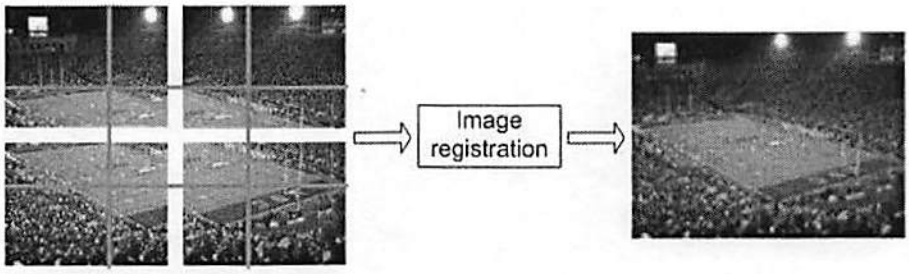
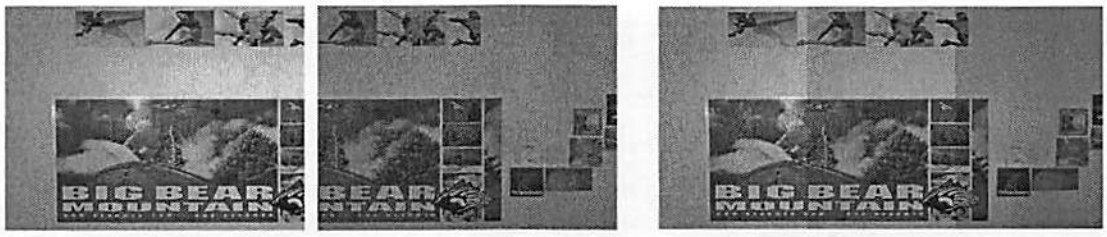


Figure 2.3: Illustration of image registration and mosaicking.



(a) (b)

Figure 2.4: Illustration of color matching and compensation: (a) two input color images and (b) the image mosaic without color adjustment.

- **Color Matching and Compensation**

  The calibration of different cameras may be different so that their color preference may vary. This could result in different color tones of images. As shown in Fig. 2.4 (b), the stitched image mosaic looks unpleasant since it contains two apparent seam lines around the image boundary. The object of this process is to adjust the pixel values of two images so that the color tones will become similar to each other. As a consequence, the seam lines in the stitched image will be eliminated.

2.2 Review of Traditional Image Registration Techniques

When input image/video contents are taken from different cameras with a different viewpoint, sampling time and sensor, image registration is needed to integrate these image/video tiles together. Most traditional image registration techniques are developed

in the pixel domain. They consist of two major steps: feature detection and feature matching, which will be discussed in detail in this section.

2.2.1 Feature Detection

The main task of feature detection is to extract salient features such as region, line and point features as explained below.

- Region features

  Examples include lakes [18], forest [44], or any closed-boundary areas. They are detected by segmentation, which is done iteratively along the registration process.

- Line features

  Examples include object contours [34], line segments [53], which can be extracted by many methods. The standard ones are Canny edge detector and an edge detector based on the Laplacian of Gaussian.

- Point features

  Examples include line intersections [52], line ends and centroids of closed-boundary regions [19], [35], and corners [54], [55]. They are usually determined at positions that have a high variance such as local extrema of the wavelet transform or a curvature.

Some feature extraction methods are based on the information provided by the first- or second-order derivatives while others investigate the image behavior around corners. The specific features to use may vary according to image contents and applications. However, all of them have something in common. That is, they are locally unique, distributed over

the image, and easy for detection. Once the feature information is available, the next step is to find the optimal correspondence of features between image tiles.

2.2.2 Matching of Image Areas and Features

Feature (or image) matching is a process to determine the relationship between similar objects contained in different images (or between individual images) by finding spatial relations among extracted features. There are several ways to define the similarity or the difference measure for image/feature pairs.

- **Cross-correlation**

  The cross-correlation between images I_1 and I_2 is defined as

$$CC(i,j) = \frac{\sum (I_1(i,j) - E(I_1))(I_2(i,j) - E(I_2))}{\sqrt{\sum (I_1(i,j) - E(I_1))^2}\sqrt{\sum (I_2(i,j) - E(I_2))^2}}. \tag{2.1}$$

  where $I_1(i,j)$ and $I_2(i,j)$ are intensity values of two areas under alignment. The correlation function is used to determine the degree of closeness. To be more specific, it computes the cross-correlation of intensities of a certain region of input images to find the best match.

- **Fourier transform**

  Fourier transform converts an image from the space domain to the frequency domain. The cross-power spectrum of two images is defined by

$$\frac{F_1(\omega_x,\omega_y)F_2{}^*(\omega_x,\omega_y)}{\mid F_1(\omega_x,\omega_y)F_2{}^*(\omega_x,\omega_y) \mid} = e^{\omega_x dx + \omega_y dy} \tag{2.2}$$

where dx and dy are the displacement parameters. The displacement is determined by the peak of the cross-power spectrum.

- **Mutual information**

  The mutual information is defined as

$$I(I_1, I_2) = H(I_2) - H(I_2 \mid I_1) = H(I_1) + H(I_2) - H(I_1, I_2), \tag{2.3}$$

  where $H(I) = -\sum_{i \in I} p(i) \log p(i)$ is the entropy of source I. and $p(i)$ is the probability function of i. The goal here is to maximize the value of mutual information, $I(I_1, I_2)$.

- **Norms of image difference**

  The sum of absolute differences (SAD) and the sum of squared difference (SSD) of image pixels are two commonly used metrics. They are defined as

$$SAD = \sum_{i,j} | \, (I_1(i,j) - I_2(i,j)) \, |, \tag{2.4}$$

$$SSD = \sum_{i,j} (I_1(i,j) - I_2(i,j))^2. \tag{2.5}$$

- **Hausdorff distance**

  The Hausdorff distance is defined as

$$H(A, B) = max\{h(A, B), h(B, A))\} \tag{2.6}$$

where A and B are two point sets, $h(A,B) = \sup_{a \in A} \inf_{b \in B} \| a - b \|$, and where $\| \cdot \|$ is the Euclidean norm of a and b. It was reported in [22] that it outperforms the cross-correlation method.

The feature matching process can be also treated as an optimization problem, which maximizes the similarity measures (*e.g.* cross correlation, Fourier transform and mutual information) or minimizes the difference measures (*e.g.* norms of image difference and the Hausdorff distance). Several solutions have been proposed to solve this optimization problem, including the Gaussian-Newton minimization, the gradient descent optimization, and the Levenberg-Marquart optimization.

Other than the measures mentioned above, some researchers adopt the multi-scale approach which registers images from the coarse to the fine scale. The wavelet decomposition is a representative of this hierarchical method, where the image is divided iteratively into four subbands of different frequencies.

2.2.3 Geometric Image Transforms

Another approach to solve the registration problem is finding a geometric transform between two images. Geometric transforms may include the rigid, affine, projective, perspective and polynomial transforms. They are explained below.

- **Affine transform**

  An affine transform is usually composed of translation, rotation and scaling. It can be expressed in a general form as

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \tag{2.7}$$

- **Perspective transform**

  The perspective transform is used to model the effect of projecting a 3D scene onto a 2D image plane. Consider that the Cartesian coordinates in the 3D space, denoted by (x, y, z). Then, its corresponding coordinates in the image plane can be expressed by

$$x' = \frac{-fx}{z-f}, \qquad y' = \frac{-fy}{z-f}, \tag{2.8}$$

  where f is the focal length of the camera.

- **Projective transform**

  If a scene plane is not parallel to the image plane, the scene is mapped onto the image plane through the following projective transform:

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}}, \quad y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}}, \tag{2.9}$$

  where a_{ij} are constants.

- **Polynomial transform**

  The polynomial transform is adopted for the case where the geometric model of the camera is unknown. The transformation can be written as the following form:

$$x' = \sum_{i=0}^{m}\sum_{j=0}^{i} a_{ij}x^i y^{j-i}, \quad y' = \sum_{i=0}^{m}\sum_{j=0}^{i} b_{ij}x^i y^{j-i}. \tag{2.10}$$

2.2.4 Optical Flow

The optical flow approach has been recently proposed for video registration with good performance [3]. Let $I(x, y, t)$ be a function of the image intensity. Then, the behavior of the neighborhood centered at position (x, y) over a short period of time can be expressed as

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \cdots, \tag{2.11}$$

where higher order derivatives are assumed to be negligible. If point (x, y) moves to a new position $(x+dx, y+dy)$ over a period of time dt, we have $I(x+dx, y+dy, t+dt) = I(x, y, t)$. Then, (2.11) can be rewritten as

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \approx 0. \tag{2.12}$$

Dividing both sides of Eq. (2.12) by dt and letting $\frac{dx}{dt} = u$, $\frac{dy}{dt} = v$ leads to

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \tag{2.13}$$

which is called the optical flow equation. It represents the intensity changes along x and y directions and along time t. The problem is an ill-posed one and its solution demands some additional constraints. Once the constraints are added, the problem can be solve by the Lagrange multiplier method.

Generally speaking, automatic image registration is still an open problem. Researchers still continue to look for better algorithms with robust performance in various application environments.

2.3 Review of Traditional Super Resolution and Image Enhancement

The super resolution problem is formulated below. For a more detailed discussion on the super resolution problem, we refer to [4] and [49].

Let $\{\underline{Y}_k\}_{k=1}^N$ and $\underline{X}$ be the set of N low resolution input images and the desired high resolution image, respectively. Then, by taking various degradation effects into consideration, the relationship between $\underline{Y}_k$ and $\underline{X}$ can be written as

$$\underline{Y}_k = D_k B_k W_k \underline{X} + N_k \qquad k = 1, \cdots, N, \tag{2.14}$$

where W_k is the warping matrix, B_k the blur matrix, D_k the subsampling matrix and N_k the noise. This is called the image observation model [39] and shown in Fig. 2.5. Note that, for most cases, N_k is assumed to be white Gaussian noise with correlation function $E\{N_k {N_k}^T\} = \sigma^2 I$. The super-resolution problem is to recover $\underline{X}$ based on observations $\underline{Y}_k$ with $1 \leq k \leq N$.

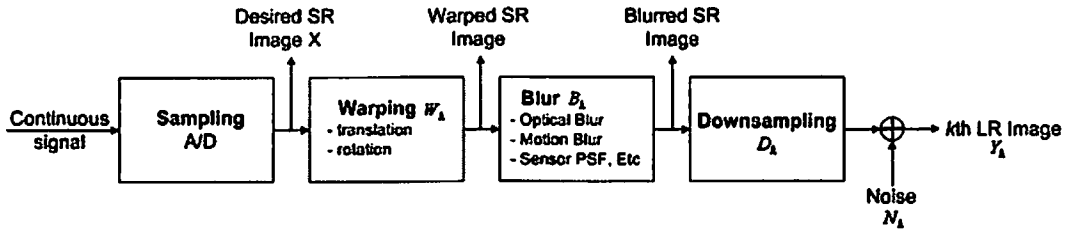


Figure 2.5: The observation model for super resolution.

Although many existing super-resolution techniques can provide high quality output, most of them deal with raw video in the pixel domain rather than compressed video such as MPEG in the DCT domain. Even some of them adopt the frequency-domain information, they still demand the manipulation in the pixel (or space) domain.

2.3.1 Spatial-domain Algorithms

Generally speaking, spatial-domain algorithms can be categorized into following types based on the underlying techniques: interpolation, iterated back projection (IBP), stochastic reconstruction methods, set theoretic reconstruction methods, hybrid ML/MAP/POCS methods and optimal adaptive filtering. Each of them will be discussed in the following.

- **Interpolation**

  Interpolation is the most intuitive method to enhance the resolution of an image. The most commonly used one is the bilinear interpolation. There are however other interpolation schemes available. For example, Landweber algorithm [27] was used by Komatsu *et al.* [26] and Shah *et al.* [45] and a weighted nearest neighbor interpolation was adopted by Alam *et al.* [1]. A wavelet-based algorithm was introduced by Nguyen *et al.* [36] to deal with interlaced two-dimensional data. Generally speaking, interpolation operations are easy to implement. However, they are not related to the

observation model. Important issues such as image degradation due to optical blur, motion blur and noise, cannot be well treated by this approach.

- **Iterated Backprojection**

Let $\tilde{\mathbf{x}}$, $\tilde{\mathbf{Y}}$ and $\mathbf{H}$ be the estimates of the desired super-resolution image, the estimated LR image and the image observation model. Then, we have $\tilde{\mathbf{Y}} = \mathbf{H}\tilde{\mathbf{x}}$. The iterated backprojection (IBP) is a process that backprojects the error between the k^{th} estimated LR image $\tilde{\mathbf{Y}}^{(k)}$ and the observed LR image $\mathbf{Y}$ by a matrix denoted by $\mathbf{H}^{\mathbf{BP}}$. The generalized iterative equation can be written as

$$\begin{aligned} \tilde{\mathbf{x}}^{(k+1)} &= \tilde{\mathbf{x}}^{(k)} + \mathbf{H}^{\mathbf{BP}}(\mathbf{Y} - \tilde{\mathbf{Y}}^{(j)}) \\ &= \tilde{\mathbf{x}}^{(k)} + \mathbf{H}^{\mathbf{BP}}(\mathbf{Y} - \mathbf{H}\tilde{\mathbf{x}}^{(j)}) \end{aligned} \tag{2.15}$$

The above procedure is performed iteratively until the error between the estimated and the observed ones converges. Note that this method is not applicable if there is some *a priori* information on $\mathbf{x}$ that has to be taken into account.

- **Stochastic Reconstruction**

When there is *a priori* information on $\mathbf{x}$, a stochastic method called the Bayesian approach can be adopted. As mentioned before, the image observation model is of the form: $\mathbf{Y} = \mathbf{H}\mathbf{x} + \mathbf{N}$. The *Maximum A-Posterior (MAP)* estimate of $\mathbf{x}$ can be derived by maximizing the power density function (PDF) $P\mathbf{x}|\mathbf{Y}$ as

$$\begin{aligned} \tilde{\mathbf{x}}^{(k+1)} &= arg\max P(\mathbf{x}|\mathbf{Y}) \\ &= arg\max P\{\ln P(\mathbf{Y}|\mathbf{x}) + \ln \mathbf{P}(\mathbf{x})\}, \end{aligned} \tag{2.16}$$

where $\ln P(\mathbf{Y}|\mathbf{x})$ is the log-likelihood function and $P(\mathbf{x})$ is the *a priori* density of $\mathbf{x}$. Note that $P(\mathbf{x})$ is often represented by the Markov random field (MRF) according to the local neighboring interaction model. The Huber MRF was adopted by Stevenson in [43]. The Gaussian MRF was used by Hanson *et al.* [9] and Hardie *et al.* [17], where the latter takes not only the global motion information but also the PSF of the sensor/optical system into consideration.

Another class of stochastic reconstruction methods is based on the *maximum likelihood (ML)* formulation for registration, interpolation and restoration [48], [49], [11]. Katsaggelos [49] used this scheme to estimate the effect of sub-pixel shifts and noise variance and the super-resolution image at the same time. The problem was solved by a method called the *expectation-maximization (EM)* algorithm. Since the *a prior* information plays an important role in the ill-posed super-resolution problem, the ML algorithm is less preferable than the MAP estimation since the ML algorithm may not incorporate all prior knowledge properly.

Generally speaking, stochastic methods including both MAP and ML estimates provide a powerful suit of tools in the modeling of noise and the stochastic nature of underlying image and video.

- **Set Theoretic Reconstruction**

The projection onto the convex set (POCS) is one of the prominent methods for solving the super-resolution problem. This idea was first introduced by Stark and Oskoui [37]. The *a priori* knowledge about the solution can be treated as imposing constraints on the solution so that it is an element of the intersection of several

convex sets denoted by C_i, $i = 1, \cdots k$, where each C_i consists of vectors that satisfy certain properties.

Given point $\mathbf{x}$ in the space, P_i is the projection that projects $\mathbf{x}$ onto the closest point of set C_i. After applying the process iteratively, *i.e.* projecting point $\mathbf{x}$ onto all constraint sets, we have $\mathbf{x}^{(n+1)} = P_k P_{k-1} \cdots P_2 P_1 \mathbf{x}$ to fall in the intersection set, $C_I = \cap_{i=1}^{m} C_i$, which meets all constraints. Note that the closedness and the convexity of the constraint sets only guarantee the convergence of the iteration but not the uniqueness of the solution. Actually, the final solution highly depends on the initial guess. The POCS method is popular due to its simplicity, flexibility of the spatial domain observation model and the ease of incorporating *a priori* information.

- **Hybrid ML/MAP/POCS Methods**

To combine the advantages of stochastic reconstruction methods and POCS, a hybrid method was proposed in [43], [12]. If there are M constraints, the optimization can be modified as

$$\begin{aligned} &\text{Minimize } \epsilon^2 = [\mathbf{y_k} - \mathbf{H_k x}]^T \mathbf{R_n}^{-1} [\mathbf{y_k} - \mathbf{H_k x}] + \alpha [\mathbf{S_x}]^{\mathbf{T}} \mathbf{V} [\mathbf{S_x}] \\ &\text{subject to } \mathbf{x} \in C_k, \quad 1 \leq k \leq M, \end{aligned} \tag{2.17}$$

where $\mathbf{R_n}$ is the autocorrelation matrix of noise, $\mathbf{S}$ is the Laplacian operator, $\mathbf{V}$ is the weighting matrix to control the smoothing strength of each pixel, and C_k is the constraint set. This hybrid method benefits from the optimal estimates of stochastic reconstruction methods and the flexibility of including linear or nonlinear *a priori*

information of POCS. Thus, it is applicable under a more generic setting with good performance.

- **Adaptive Filtering Approach**

  The inverse filtering technique can also be used in solving the super-resolution problem. Jacquemod *et al.* [23] proposed a deconvolution process for observed images obtained through sub-pixel translation motion. A linear minimum mean squared error (LMMSE) algorithm, which can be viewed as a motion compensated multi-frame Wiener filter, was proposed by Erdem *et al.* [13] to process images with a spatial blur and additive noise. In addition to the Wiener filter, the Kalman filter was also adopted for super-resolution reconstruction in [40], [10]. This computationally efficient scheme can deal with images degraded by the spatially-varying blur. However, it cannot handle nonlinear modeling constraints effectively.

2.3.2 Frequency-domain Algorithms

To solve the super-resolution problem, the frequency-domain method [50] utilizes the shift property of the Fourier transform, the relationship between the continuous Fourier transform (CFT) and the discrete Fourier transform (DFT) and the assumption that the underlying image is band-limited. Although there are disadvantages associated with the frequency-domain approach, it is still computationally attractive while degraded images only have sub-pixel global translation motion. The frequency-domain approach [5] applied to the super-resolution problem is reviewed below.

Let $f(x,y)$ denote a continuous scene. Consider the following R shifted images

$$f_r(x,y) = f(x + \Delta x_r, y + \Delta y_r), \quad r = 1, 2, \cdots R. \tag{2.18}$$

Their continuous transforms are denoted by $F(u,v)$ and $F_r(u,v)$, respectively. By applying the Fourier transform to (2.18), we obtain

$$F_r(u,v) = \exp^{j2\pi(\Delta x_r + \Delta y_r v)} F(u,v). \tag{2.19}$$

The observed images, $y_r[m_1, m_2]$, can be obtained by sampling the original image. That is, we have

$$y_r[m_1, m_2] = f(mT_x + \Delta x_r, nT_Y + \Delta y_r), \quad m, n = 0, 1, \cdots, M-1.$$

Let $Y_r[k,l]$ be the DFT of y_r, $r = 1, 2, \cdots R$. Then, we have

$$Y_r[k,l] = \alpha \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} F_r\left(\frac{k}{MT_x} + pf_{s_x}, \frac{1}{NT_y} + qf_{s_y}\right), \quad r = 1, 2, \cdots R, \tag{2.20}$$

where $f_{s_x} = 1/T_x$ and $f_{s_y} = 1/T_y$ are the sampling rates along the horizontal and vertical directions, respectively. Based on (2.19), Eq. (2.20) can be rewritten in form of

$$\mathbf{Y} = \mathbf{\Phi F}, \tag{2.21}$$

where $\mathbf{Y}$ is the DFT coefficient vector with elements $Y_r[k,l]$, $r = 1, 2, \cdots R$, F is the vector consisting of samples of the CFT of the high resolution image, and Φ is the matrix that

models the relationship between Y and F. Then, the solution to the super-resolution problem can be obtained by solving a linear system for F and then applying the inverse DFT to the resulting F to reconstruct the space-domain image f. This frequency domain solution procedure saves a lot of computation.

However, the assumptions made here are not realistic since the optical point spread function (PSF) as well as the observation noise are not considered. Some extensions of [4] have been made by Kim *et al.* [24] and Tekalp *et al.* [47], who took PSF and noise into consideration and solved the problem by the least squares method. Later on, the recursive least squares solution for (2.21) was proposed by Bose *et al.* [6], where the problem is modified to minimize

$$\| \Phi F - Y \|^2 + \lambda \| F - \mathbf{c} \|^2, \tag{2.22}$$

and where $\mathbf{c}$ is an approximation to the desired solution. The solution to this problem becomes

$$\tilde{\mathbf{F}} = (\Phi^T \Phi + \lambda \mathbf{I})^{-1} (\Phi^T Y + \lambda \mathbf{c}), \tag{2.23}$$

which can be solved iteratively rather than directly via matrix inversion. With a fast convergence rate, the computational complexity of an iterative method can be reduced dramatically. Kim *et al.* [6] proposed a recursive total least squares method to solve a problem where errors appear in observations as well as the system matrix. Then, the observation can be expressed as

$$\mathbf{Y} = [\Phi + \mathbf{E}]\mathbf{F} + \mathbf{N}, \tag{2.24}$$

where $\mathbf{E}$ is the motion estimation error in $\mathbf{\Phi}$, and $\mathbf{N}$ is additive noise. The problem can be further converted to a constrained optimization problem as

$$\begin{aligned} &\text{Minimize } \| [\mathbf{N}\vdots\mathbf{E}] \|_F, \\ &\text{Subject to } \mathbf{Y} - \mathbf{N} = [\mathbf{\Phi} + \mathbf{E}]\mathbf{F}. \end{aligned} \tag{2.25}$$

If there exists a simple expression of the relationship between the low-resolution and high-resolution images, frequency-domain methods are computationally attractive. However, these methods can handle global translational motion and spatial invariant degradation only. It is difficult for them to deal with generic degradation models and to incorporate *a priori* information. These are their main limitations.

2.4 Challenges of Coded Video Mosaicking and Research Objectives

Even though image/video mosaicking has been studied for decades, most techniques have been developed based on raw video data (or uncoded video). Thus, most algorithms are implemented in the space-time domain (or the image pixel domain). In this research, we consider mosaicking of coded video since each individual captured video is coded before its transmission. If we perform image/video mosaicking in the raw video domain, it demands image/video decoding first. For example, it will involve tedious inverse/forward DCT when the input video is in the compressed format such as motion JPEG and MPEG. They are not suitable for implementation in embedded real-time systems due to the heavy computation and large memory needed in this process. Thus, it is desirable to study

image/video mosaicking in the compressed domain directly, which is one of the main tasks of this research.

The super resolution problem has been studied for a long while. However, all existing algorithms have been designed for raw video inputs. For multiple coded image sequences, how to get super resolution video based on coded video in the DCT domain (without decoding them back to the raw video domain fully) to save computation as well as storage is clearly a great challenge. Some information in the coded video domain, such as DCT coefficients, quantization step sizes, motion vectors or even the residuals, could help improve the performance of super-resolution outputs. However, a super resolution method fully in the DCT domain could be very difficult due to the limitation of the frequency domain methods as discussed in the last subsection. Thus, to develop the super-resolution technique for multiple MPEG video sequences, our initial goal is to develop a hybrid method that utilizes raw as well as coded video data to produce a high-resolution MPEG video output. However, in order to save the computational cost, we prefer to perform operations in the compressed domain as much as possible. The pixel domain process will be considered only when it is absolutely needed.

DCT provides a powerful tool for energy compaction (by removing spatial redundancy of the underlying image), and it is widely used in image coding standards such as JPEG and MPEG. Thus, we study image/video registration, color matching, and super resolution techniques based on multiple coded video clips in the DCT domain. Besides, motion vectors can provide auxiliary temporal information for image alignment. The proposed system is illustrated in Fig. 2.6.

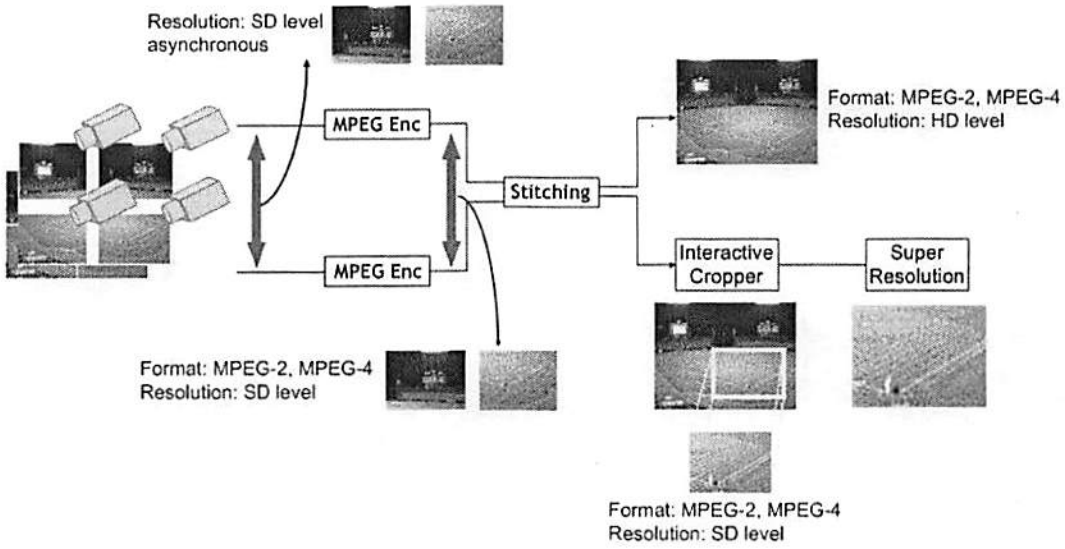


Figure 2.6: Illustration of the proposed video mosaicking system.

The ultimate object of this research is to develop an efficient system to generate a mosaic image/video using several images/videos captured by different sensors under different conditions. Suppose there are few input video sequences of SD level produced by different cameras and then passing through the MPEG encorders to generate compressed MPEG format streams. A stitching scheme is performed mostly in the compressed domain without going through the conversion to the spatial domain. Note that there are some assumptions made in the proposed system. For example, the temporal synchronization parameters are well calibrated and the radial distortions caused by various focal length are also compensated in advance. Then, there are two major research issues remaining: namely, color matching and image/video registration. They need to be addressed to compensate the discrepancy between two image/video inputs to make one high quality stitched output.

In our research, most of process are performed in the DCT domain so that a high resolution image sequence can be obtained from several low resolution image sequences

via video mosaicking. Furthermore, a post-processing technique called super resolution can be applied to any region-of-interest or even the whole image for the highlight purpose.

Chapter 3

Color Matching and Compensation of Coded Images

A video source captured by a camera usually has its unique color preference and response to the light. Consequently, the brightness and color of different video sources may vary significantly. Even though images are perfectly stitched geometrically, apparent seam lines may still exist between video tiles [35]. To make the image/video mosaic look more natural, it is important to find a way to adjust the color tones of image/video inputs as close as possible.

Several ideas in comparing the color similarity between two images have been studied. For example, a structured light approach was proposed by Tsukada and Tajima [51]. More recently, a new technique was proposed by Hu and Mojsilovic [21] to extract relevant colors, where a new color distance measure was used to assure the optimal matching of different colors from two images. Most of previous color matching algorithms are conducted in the spatial domain, which demands a larger amount of computation. Since many video sources are encoded using the motion-compensated predictive coding technique such as

the MPEG and H.26x coding standards, it is preferred that the task of color matching and correction is done in the DCT transform domain (rather than in the pixel domain) in the video mosaic authoring process.

Under the simplifying assumption that all other differences have been compensated, we examine and compare various color-matching techniques in the pixel and the DCT (Discrete Cosine Transform) domains in this chapter. Algorithms in the pixel domain as well as the transform domain for color difference compensation and seam line removal among video tiles are proposed. They are compared in terms of visual quality. It will be demonstrated by experimental results that the transform-domain algorithm can achieve good quality of composition while dramatically reducing the computational burden of decoding/encoding.

The overlapping region of two images can be determined in the image/video registration process, which will be described in Chapter 4. Here, it is assumed that the overlapping region of two adjacent images has I columns. For these I columns, we perform some manipulation to make each color component of two images to share a similar range of values. Two approaches are considered. They are the histogram-based approach and the polynomial-based contrast stretching approach, which will be described in Sec. 3.2 and Sec. 3.3, respectively.

3.1 Pre-processing via White Balancing

Before the compensation of the color difference between two images, a pre-processing called white balancing may be needed to correct the color tone of each image individually. White balancing is the process of adjusting image colors under different illumination conditions

so that the color bias of each image can be removed. In other words, an object that appears in white in human eyes are rendered white in the photo. It is not difficult for our eyes to determine the white color under a different light condition automatically but it is a difficult task for cameras to do so. A technique called the Gray World Assumption (GWA) has been developed to achieve this goal and will be reviewed below.

3.1.1 Fundamentals of Gray World Assumption (GWA)

The GWA states that a given image is assumed to have a sufficient amount of color variations. In other words, the average value of the R, G and B components of an image should average out to a common gray value. Under this assumption, the three channels are adjusted individually but the adjustment ratios for each channel should be kept the same for all pixels. This procedure can be done detailed as follows.

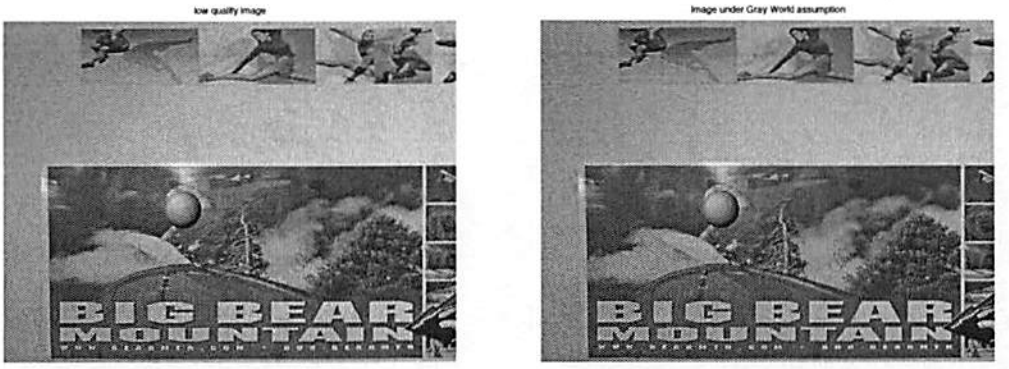
Consider R, G, and B three color components. The first step is that the mean of each component as well as the total mean are calculated and denoted by m_R, m_G, m_B and m_{RGB}, respectively. The ratio of each component is then defined as

$$r_R = \frac{m_{RGB}}{m_R}; \quad r_G = \frac{m_{RGB}}{m_G}; \quad r_B = \frac{m_{RGB}}{m_B}. \tag{3.1}$$

According to these ratios, the color components R, G, and B can be adjusted by

$$R_{GWA} = r_R \cdot R; \quad G_{GWA} = r_G \cdot G; \quad B_{GWA} = r_B \cdot B. \tag{3.2}$$

An example of GWA is shown in Fig. 3.1. We see that the color tone has been compensated and the output image quality has been improved.



(a) (b)

Figure 3.1: Experimental results of applying the Gray World Assumption to all image pixels: (a) the input image and (b) the output image.

3.1.2 White Balancing in DCT Domain

By examining HSI components and the corresponding histograms of the two images in Fig. 3.1, we see that the main difference lies in the H component while there is not much difference in the S and I components. It indicates that GWA focuses more on the adjustment of the H component and less on that of the S and the I components. If we examine GWA in the YCbCr space, we see that it has more impacts on the chrominance components, *i.e.* Cb and Cr. Two different cases are considered below to verify the idea of performing GWA on Cb and Cr components.

For the first case, an image with lower saturation and lower intensity is shown in Fig. 3.2(a) and the corresponding output of the GWA enhanced image is shown in Fig. 3.2(b). We see that the yellowish color tone has been corrected to some degree but the output image quality is poor due to the low intensity.

For the second case, an image with a different hue component but the same two other channels is shown in Fig. 3.1(a) and the corresponding output of the GWA enhanced image is shown in Fig. 3.1(b). The quality of the output image is significantly enhanced.

Finally, by comparing Fig. 3.1 (b) and Fig. 3.2(b), we see that GWA works better for the hue adjustment (color tone manipulation) but cannot do much for the adjustment of the saturation and intensity components.

(a) (b)

Figure 3.2: Experimental results of applying GWA to an image with low saturation and low intensity components.

3.1.3 Experimental Results

A preliminary experiment of performing GWA in the DCT domain has been conducted to verify the proposed idea of improving color quality of an image. As shown in Fig. 3.3, we find that the result is similar to the one in Fig. 3.1(b). In other words, GWA is applicable to the chrominance components to compensate the color tone of an image successfully. Note the GWA works well under the assumption that the underlying image has a sufficient amount of color variations. Thus, if the image content is dominated by a certain color, the algorithm may fail to provide a high quality output image. Another color processing approach is proposed in the next section to deal with this situation.

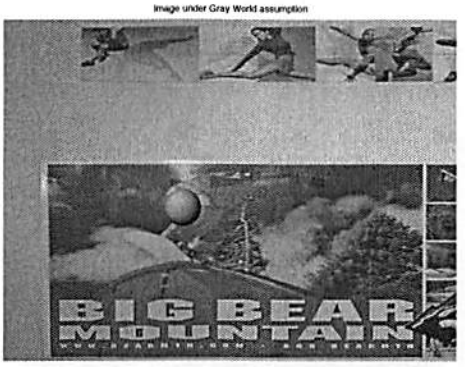


Figure 3.3: Experimental results of applying GWA in the DCT domain.

3.2 Histogram Matching

In this section, we consider the problem of adjusting histograms of two images to eliminate the seam lines around boundaries in an image mosaic using the histogram matching method. The color associated with each pixel can be represented using the RGB or the YCbCr color coordinates. In the following discussion, we focus on the histogram adjustment for any single color component (*i.e.* a monochrome image). The same process can be applied to the three color components separately.

3.2.1 Fundamentals of Histogram Matching Technique

The histogram of an image is obtained by choosing a bin size, which is usually 256 or fewer since we adopt the 8-bit representation for a monochrome image, and counting the number of pixels with values belonging to each bin. Finally, we may divide the number of each bin by the total number of pixels for the normalization purpose. If we treat an image as a source and its gray level value a random variable, then the histogram can be viewed as the probability density function (pdf).

By histogram matching, we refer to a process that adjusts the histogram of an image to be similar to the desired one. We use I_1, H_{I_1} and H_d to denote an input monochrome image, its histogram and the histogram of the desired output monochrome image, respectively. Our goal is to find the mapping F so that $H_{F(I_1)} = H_d$. To perform histogram matching, we need the cascade of the following two steps.

- Step 1: Map from I_1 to I' using the cumulative distribution function (cdf) of input image I_1, which is denoted by F_1, and
- Step 2: Map from I' to $F(I_1)$ using the inverse cumulative distribution function (icdf) with respect to the desired histogram H_d, which is denoted by F_2.

Please note that the cdf of an image is computed from its histogram by summing up successive bin counts from 0. The cdf is a function that maps the interval [0,256] to [0,1] while the inverse cdf is another function that maps [0,1] back to [0,256]. Thus, the histogram equalization can be written as

$$F(I_1) = F_2[F_1(I_1)].$$

Fig. 3.4 shows an example, where the histograms of three components (RGB) of an image are adjusted by applying the histogram matching technique. The three sub-figures in the top row of (a) are histograms of the R, G, B color components of the first original input and the bottom row are those of the second original input. The histograms of first and second output images after histogram matching are shown in the top and bottom rows of (b), respectively.

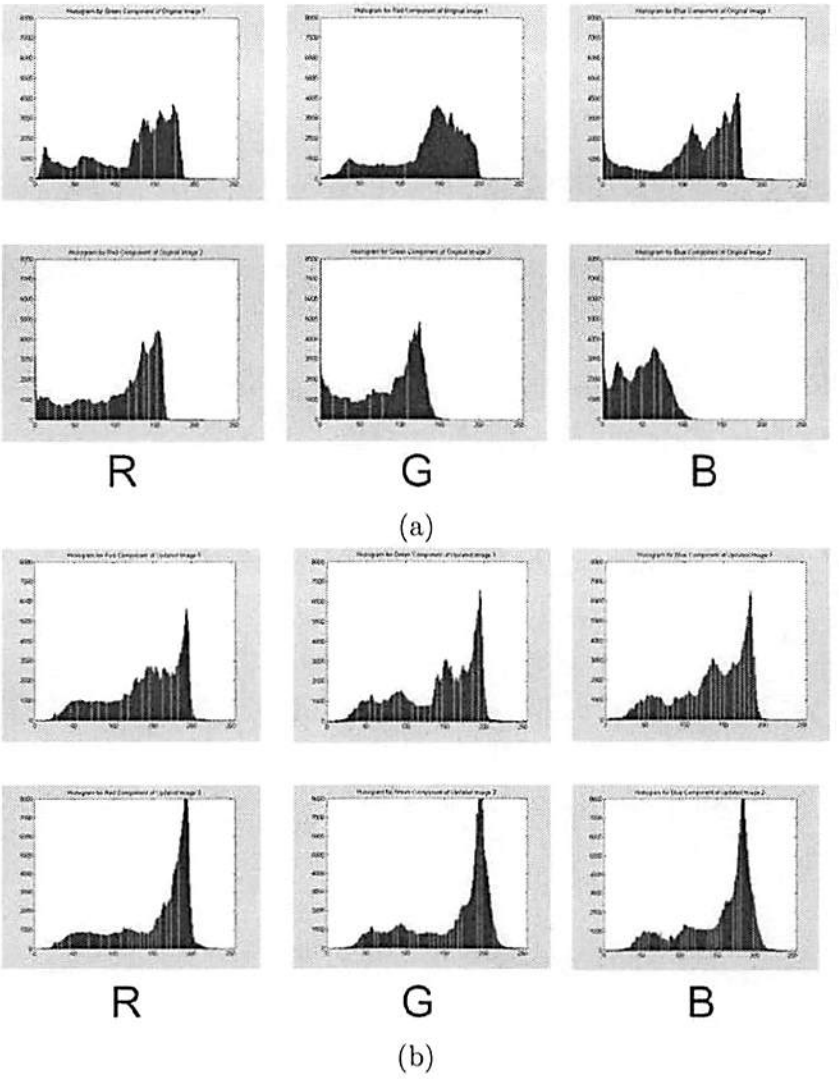


Figure 3.4: The histograms of three components of an image: (a) before and (b) after histogram matching.

3.2.2 Pixel-domain Color Adjustment

Since it is in the pixel domain, the image color is typically represented by its red (R), green (G), and blue (B) components. Take the R component for example. Let $\tilde{R}_1(x, y)$ and $\tilde{R}_2(x, y)$ be two 2-D sequences of the red component in the overlapping region for image 1 and image 2, respectively. We first compute the mean $m_R(x, y)$ of the corresponding pixels in these two sequences. Then, we can generate histograms for three sequences separately: $\tilde{R}_1(x, y)$ and $\tilde{R}_2(x, y)$ and the mean sequence $m_R(x, y)$. Afterwards, we can define a mapping table, L_{R_1}, for pixels of image 1 in the overlapping region so that the histogram of $\tilde{R}_1(x, y)$ alone can be converted to that of the mean sequence. Then, by adopting the same mapping rule, we are able to update the R components of all other pixel values for image 1. The same procedure can be applied to image 2. After the R, G, and B components of both images are properly updated with the above procedure, these two new images should have the same color tone in the common (*i.e.* overlapping) region as well as the two disjoint regions.

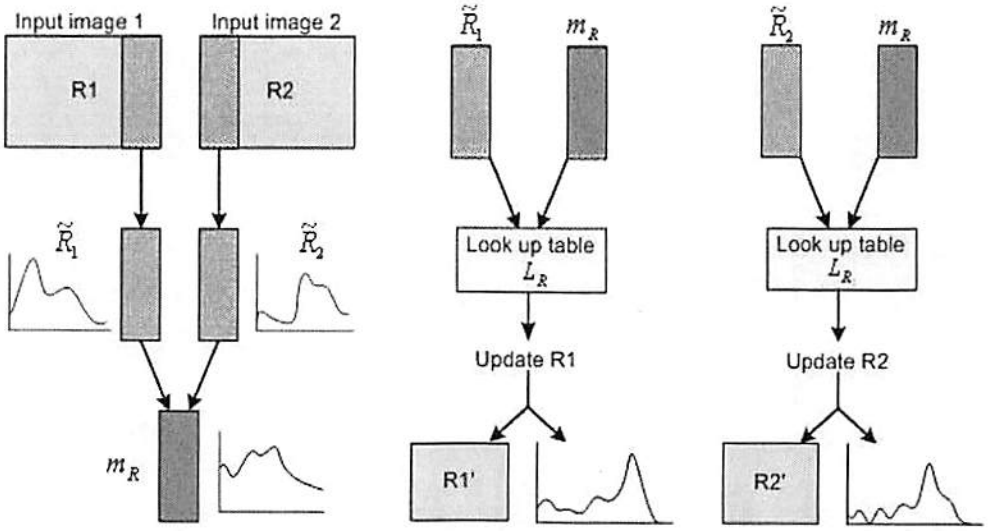


Figure 3.5: The block diagram of histogram matching in the pixel domain.

3.2.3 DCT-domain Color Adjustment

Instead of dealing with RGB components, we manipulate the DC values of YCbCr for color adjustment when performing color adjustment in the DCT domain, since image/video coding is usually done in the YCbCr domain.

1. Basic scheme for exactly matched block locations

   For the DCT domain processing, we first consider a simplified case where the DCT blocks from image 1 and image 2 have the exact match as shown in Fig. 3.6(a). Then, we perform the color matching process by adjusting the Y, Cb, Cr values of the DC coefficient of each DCT block. For each color component of the DC coefficient, we can adopt the histogram matching method which is described in Sec. 3.2 for color adjustment.

2. Modified scheme for blocks with displacement

   Next, let us consider a more complicated case, where the DCT blocks from image 1 and image 2 are offset by an amount of m and n pixels along the horizontal and the vertical directions, respectively, where m and n are in the range of 0 and 8. Then, we can proceed with the following three steps.

   - Step 1: Once the spatial displacement (m,n) is known, we use the bilinear interpolation to interpolate the DC components of four DCT blocks from image 1, i.e. a, b, c, and d shown in Fig. 3.7, that surround the target DCT block of image 2 so that the interpolated block has the same spatial location as the target DCT block. The interpolated DC value is called the DC component of the pseudo DCT block in image 1.

- Step 2: We adjust the color values of the DC components of the DCT block in image 1 and the pseudo DCT blocks in image 2 using the algorithm presented in Sec. 3.2.

- Step 3: We use the bilinear interpolation again to interpolate the DC components of the DCT blocks of image 2 based on the DC components of their surrounding pseudo DCT blocks. This process is illustrated in Fig. 3.7.

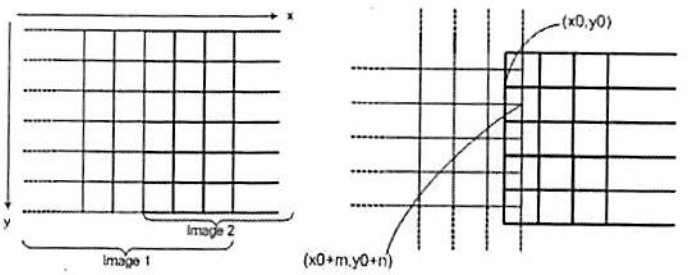


Figure 3.6: The DCT blocks from images 1 and 2 have (a) the exact matched location and (b) an offset of (m,n).

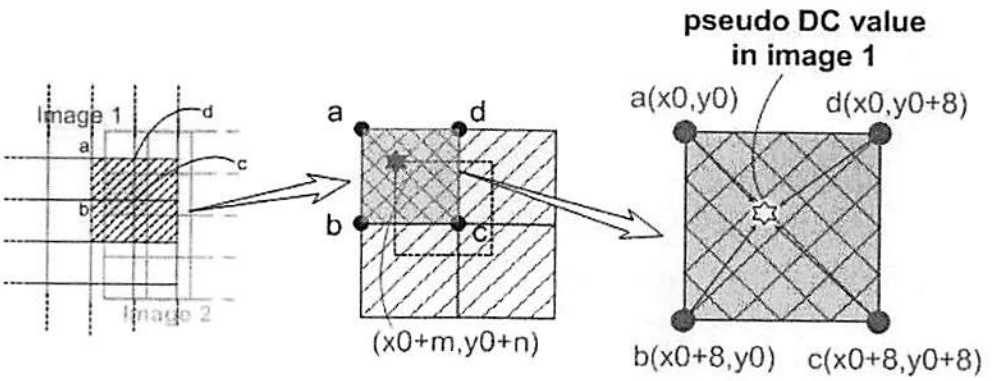


Figure 3.7: The bilinear interpolation of four DCT blocks to synthesis the pseudo DCT blocks and vice versa.

3.3 Polynomial Approximation

3.3.1 Pixel-domain Contrast Stretching

The color tone adjustment can be achieved by using contrast stretching as detailed below. First, we compute the mean value of each color component of pixels in the overlapping region. Then, we can approximate the mapping from the original color component to this newly calculated color component with a polynomial which is described in Sec. 3.3 of different orders for one input image. We have n+1 coefficients for a n-th-order polynomial. These coefficients can be determined using the least square methods for all pixels in the overlapping regions. Once these coefficients are found, they can be used to update the values in the non-overlapping region of this image. The same procedure can be applied to the other input image. In our experiment, we found that it is sufficient to have a low order polynomial such as $n = 2$. The improvement comes from a larger n is very limited. More details will be shown in Sec. 3.5.2.

3.3.2 DCT-domain Contrast Stretching

Here, we still consider two cases: a basic scheme with exactly matched block locations and a modified scheme with blocks with displacement. Note that the data sequences here are DC values of YCbCr in the overlapping region instead of RGB components. After solving the linear system for coefficients **a** and **b**, we are able to update all the other DC values in the non-overlapping region. Note that the data size is 1/64 of the one in the pixel domain since each 8×8 block has only one DC value.

3.4 Post-processing via Linear Filtering

Consider two data sequences, $\mathbf{x} = [x_0\ x_1\ \cdots\ x_n]$ and $\mathbf{y} = [y_0\ y_1\ \cdots\ y_n]$. We would like to combine them together to form a new data sequence which contains some information of the original two sequences with different weights respect to their spatial positions. In terms of mathematics, the relationship between the new sequence and the original two data sequences can be expressed as

$$\mathbf{z}(i) = \alpha \times \mathbf{x}(i) + (1 - \alpha) \times \mathbf{y}(i)\ for\ i = 0, \cdots, n. \tag{3.3}$$

where $0 \leq \alpha \leq 1$ is the weighting parameter whose values are shown in Fig. 3.8.

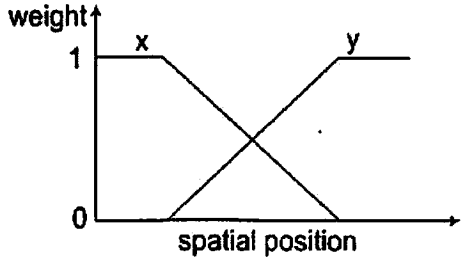


Figure 3.8: The curves of two weighting parameters used in the linear combination.

3.4.1 Pixel-domain Post-processing

After the above step, we still see two seam lines between the overlapping and the non-overlapping regions. To remove seam lines, we weigh values from images 1 and 2 and combine these weighted values into one final value. In terms of mathematics, the relationship between the new image and the original two images can be expressed as

$$\tilde{R}'(i,j) = \alpha \times \tilde{R}'_1(i,j) + (1 - \alpha) \times \tilde{R}'_2(i,j), \tag{3.4}$$

where $\tilde{R}'(i,j)$ is the new value at position (i,j) in the overlapping region of the output stitched image, $\tilde{R}'_1(i,j)$ is the updated value at position (i,j) in the overlapping region of image 1, $\tilde{R}'_2(i,j)$ is the updated value at position (i,j) in the overlapping region of image 2, and $0 \leq \alpha \leq 1$ is the weighting parameter varying according to the pixel position. In our experiment, we let the value of α increase linearly from 0 to 1 when (i,j) moves from the boundary of image 1 to the boundary of image 2 along the overlapping region.

3.4.2 DCT-domain Post-processing

After updating all DC values, the seam lines between image boundaries might still exist. To remove seam lines, we can convert the color space from the YCbCr values back to the RGB values, and apply the same method as described in Sec. 3.4.

3.5 Experimental Results

In this section, we present some preliminary experimental results with two input images as shown in Figs. 3.9(a) and (b). These two test images have a very different color tone. Besides, there is a significant overlapping region between them. It is assumed that the registration part has been done so that the two images are well aligned in the common area.

Fig. 3.10 shows the stitched image without color matching, where a simple color component averaging operation is applied in the overlapping part of two images. As we can see from the output image, there exist three regions of different color tones and two apparent seam lines between two adjacent regions.

(a) (b)

Figure 3.9: The original two input images.

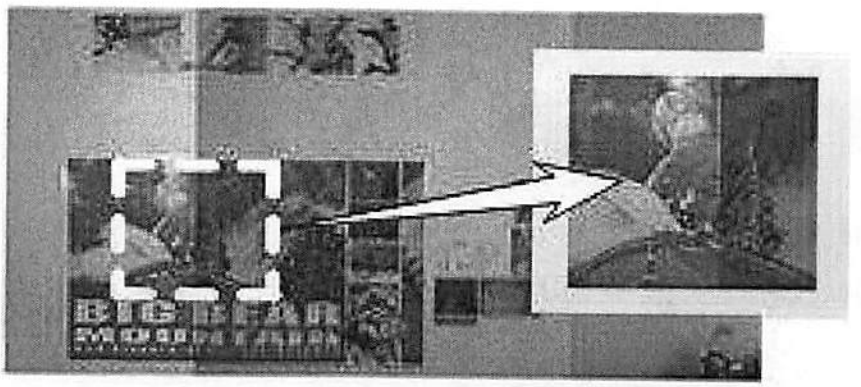


Figure 3.10: The stitched image without color matching.

3.5.1 Stitched Images After Color Matching

3.5.1.1 Histogram Matching

The stitched image using histogram matching in the pixel domain is shown in Fig. 3.11. As we can see, the color tone of the output image has been adjusted to lie in the middle of the two input images. Furthermore, the seam line has been eliminated satisfactorily. The output image looks just like a natural image with a wider angle of view.

Image stitching using the DCT domain processing based on histogram matching in the DC coefficient of DCT blocks is given in Fig. 3.12. We see the color deviation phenomenon in the non-overlapping regions since most of the pixel values fall outside the dynamic range

Figure 3.11: The output image after color adjustment in the pixel domain with histogram matching.

of $[0, 255]$ after converting the updated values of Y, Cb, and Cr back to the RGB color coordinates.

3.5.1.2 Polynomial Approximation

The result of using the polynomial-based contrast stretching in the pixel domain is shown in Fig. 3.13. The performance is similar to that of histogram matching. There are no seam lines and no color transition inside it.

The result using the polynomial-based contrast stretching as shown in Fig. 3.14 looks significantly better if compared to Fig. 3.12. The two input image have similar color tones, which are close to that of the mean color values of the overlapping region. This phenomenon can be explained below.

Figure 3.12: The output image after color adjustment in the DCT domain with histogram matching.

Since the histogram describes the overall distribution of a random variable, the more data we generate, the more accurate the description is. As to the polynomial approximation, it can be viewed as a point-wise approximation method. In our experiments, the region of the overlapping part is more than half of the original one. Thus, there is a lot of information for us to find out the relationship between these two images in the pixel domain so that both the histogram matching method and the polynomial approximation method work well. However, in the transform domain, we have one DC value for each 8×8 DCT block to find out the relationship, which is significantly less than the data in the pixel domain. Thus, the result is less robust. On the other hand, the degree of freedom in the polynomial approximation (i.e. the number of coefficients of the polynomial) is usually 3 or 4, which is still much fewer than the number of constraints in either the pixel domain or the DCT domain. Thus, the result is more robust.

Figure 3.13: The output image after color adjustment in the pixel domain with polynomial approximation.

Fig. 3.15(a) and (b) shows the stitched image using the polynomial-based contrast stretching technique in the DCT domain with exactly matched blocks and in the DCT domain with block displacement, respectively. To make the results more visible, we zoom in specific regions of the output image. As we can see, the color tone of the output image has been adjusted to lie in the middle of the two input images. Furthermore, the seam lines in Fig. 3.10 have been eliminated satisfactorily. The output image appears to be a natural image with a wider angle of view. The result for the DCT domain techniques has similar performance as the one in the pixel domain whether the block is aligned or not.

3.5.2 Performance Comparison

1. Comparison of processing time

   The comparison of processing time between the pixel and the DCT domains is shown in Table 3.1. The computation time for polynomial approximation is less than that

Figure 3.14: The output image after color adjustment in the DCT domain with polynomial approximation.

for histogram matching. As far as the memory is concerned, polynomial only needs to store three coefficients for each pair of sequences while the histogram needs an array of size at least 256 by 1. Polynomial approximation seems to be superior to histogram matching in both the computational cost and the memory requirement.

2. Comparison of MSE with different order of polynomial approaches

Figure 3.15: Stitched images with polynomial approximation in the DCT domain (a) without block displacement and (b)with block displacement.

As mentioned before, polynomial based contrast stretching has a better performance in image quality, computation complexity and memory requirement. Note that the polynomial adopted for approximation is of order 2. Theoretically, the higher the order, the more accurate the approximation. However, higher order polynomial will cost more computation complexity. Table 3.2 shows how the order would affect the performance in terms of MSE (mean squared error). As it is shown in this table, higher order approximation does improve the performance but not that obviously. Therefore, we may say that the second order polynomial approximation is sufficient to provide outputs to some satisfactory degree.

Table 3.1: Comparison of processing time (seconds) between two different domains and two different approaches.

	Pixel Domain	DCT Domain	Save
Histogram Matching	71.243 (sec)	8.623 (sec)	87.90%
Polynomial Approximation	52.135 (sec)	6.810 (sec)	86.94%
Save	26.82%	21.02%	

Table 3.2: MSE for different order polynomial approaches.

order		Y	Cb	Cr
2	image 1	5574.6	1132.4	664.8
	image 2	7001.3	1343.8	2035.5
3	image 1	5449.1	1131.9	646.9
	image 2	6980.4	1285.6	2037.3
4	image 1	5298.0	1123.8	649.5
	image 2	7011.7	1282.0	2045.6

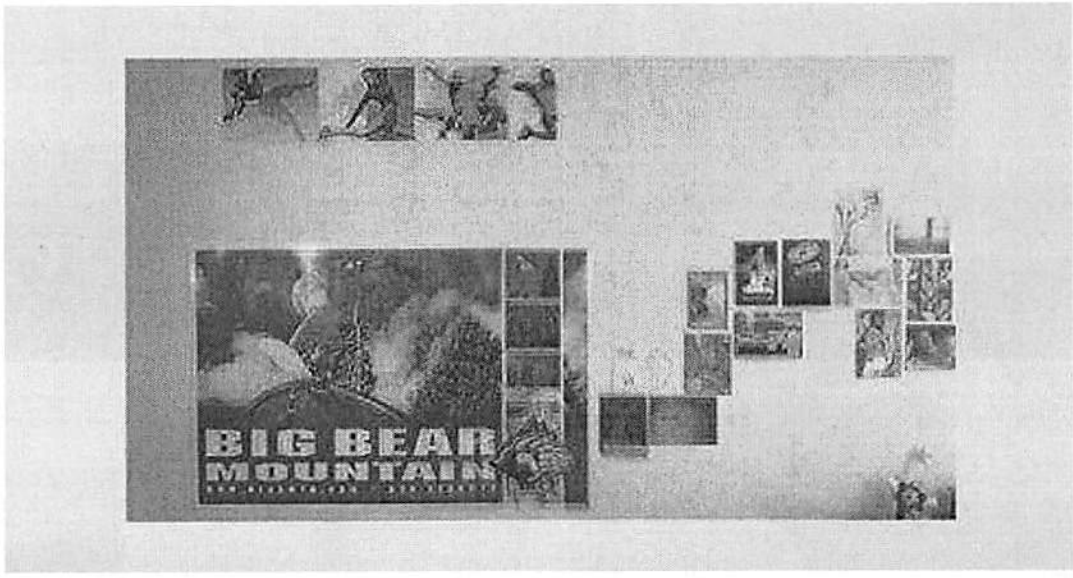
3.5.3 Other Considerations

For each 8×8 DCT block, each coefficient represents the weight of its corresponding spatial frequency, and there are 64 basis patterns. Typically, the DC value contains most information about the 8×8 block so that we only adjust DC values of two images to be the same in the DCT domain. Figs. 3.16(a) and 3.16(b) show the output images, where we try to match not only the DC values but also the first three and five AC values, respectively. The color tone is similar to the one shown in Fig. 3.14. However, we observe some artificial patterns appearing all over the whole image which degrades the quality of the output image significantly. It is not surprising since modifying DCT coefficients for each block is equivalent to changing the weights of the basis patterns. Once the proportion relationships between spatial frequencies have been modified, the output image will not be the same. This explains the occurrence of artificial patterns if we attempt to match AC values as well.

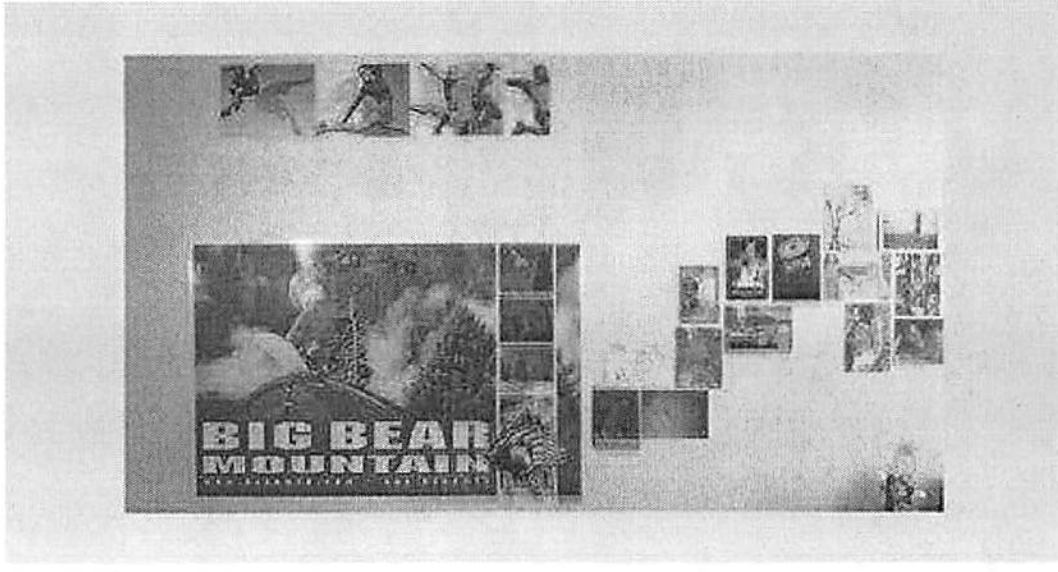
3.6 Conclusion

Several color matching algorithms that compensate the color differences from two input image sequences captured by different cameras were studied in this chapter. Some of them are conducted in the pixel domain while others are carried out in the DCT domain. It was demonstrated by experimental results that the DCT domain technique saves more than 80% computational cost as compared to the pixel domain technique while the quality of the resulting image mosaic is about the same for both approaches. It was also observed that the polynomial-based contrast stretching method has better performance than the

histogram matching method in terms of the processing time and memory requirement. Experimental results presented in this chapter are restricted to the image mosaic only. It is worthwhile to consider to generalize this technique to the video mosaic in the near future.



(a)



(b)

Figure 3.16: Image mosaic with the 2nd order polynomial in the DCT domain: (a) DC plus the first three AC values and (b) DC plus the first five values.

Chapter 4

Fast and Accurate Block-Level Registration of Coded Images

4.1 Block-level Image Registration with Edge Estimation

In this section, we study the registration of two images that contain only translation displacement in the horizontal and vertical directions. We address the problem in the DCT domain with the DC and AC values of the luminance component of each block available. We attempt to align the images to some satisfactory degree using these DCT coefficients. The proposed algorithm basically contains three steps: image segmentation for foreground extraction, edge estimation and parameter estimation.

4.1.1 Image Segmentation for Foreground Extraction

At the first step, a DC map, which contains only the DC values of the Y component of the DCT blocks, of each input image is formed. Note that the size of the DC map is 1/64 of that of the original one. To simplify the alignment process in the later stages,

(a) (b)

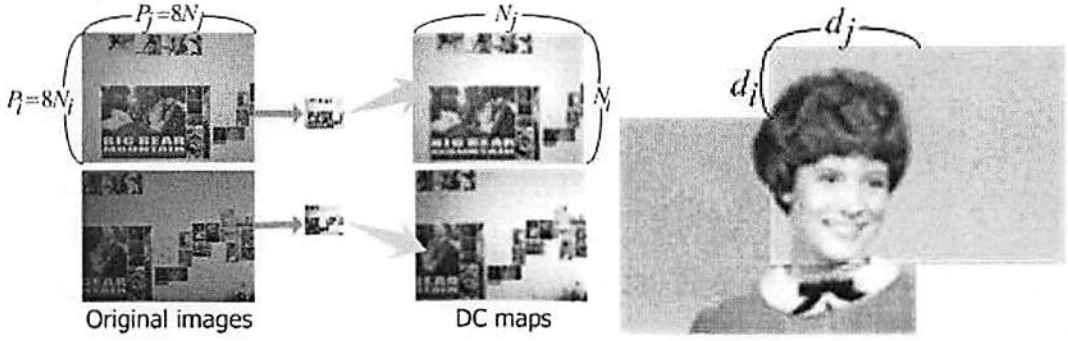
Figure 4.1: Conversion from a DC map to a binary activity map: (a)the DC map and (b)the binary activity map.

we first perform image segmentation on the DC map to extract the regions of interest. Otsu [38] proposed a method to divide the light intensity histogram into two distinct parts automatically. It is stated below and will be used for our image segmentation task. Given a histogram, we can compute its statistical properties. In particular, we use $\omega(k)$, $\mu(k)$ and $\mu_T(k)$ to represent the zeroth cumulative moment, the first cumulative moment and the mean of a bin with index k, respectively. The histogram can be split using threshold k^* such that

$$\sigma^2(k^*) = \max_{1 \leq k < L} \sigma^2(k) \tag{4.1}$$

where $\sigma^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1-\omega(k)]}$. Once the optimal value of k^* is determined, we can obtain a binary activity map B from the DC map. Ideally, the foreground (set to be 1) and the background (set to be 0) can be separated in the binary activity map. One example of converting the DC map to the binary activity map results is shown in Fig. 4.1. In Fig. 4.1 (a), we show the DC maps of two images that we intend to align. Their corresponding binary activity maps are shown in Fig. 4.1(b). Even with only the DC values, the DC maps shown in Fig. 4.1(a) still provide many fine details that make the alignment task difficult.

Suppose that the two original input images are of size $P_i \times P_j$. Then, their DC maps are of size $N_i \times N_j$, where $N_i = P_i/8$ and $N_j = P_j/8$ (as shown in Fig. 4.2(a)). Note that d_i and d_j in Fig. 4.2(b) are the displacement parameters. To search for the optimal



(a) (b)

Figure 4.2: (a)The relationship between original images and binary maps and (b) the geometrical representation for displacement.

alignment with integer block accuracy, we can compute the sum of absolute difference of the left and the right images (DC maps) with a displacement d_i and d_j:

$$\min_{d_i,d_j} \sum_{b_i,b_j} \mid L_{DC}(b_i, b_j) - R_{DC}(b_i + d_i, b_j + d_j) \mid \tag{4.2}$$

where the summation sums up all the b_i's and b_i's belonging to the overlapping region. If we do an exhaustive search based on either the DC maps or the binary activity maps, the complexity, C, will be proportional to

$$C \propto N_i \times N_j. \tag{4.3}$$

However, the computation with the binary activity maps is much faster since it takes only 0 and 1 two values. It is possible to further simplify the search if we consider the edge

information present in the foreground part of the binary activity map. This is done in the next step.

4.1.2 Edge Estimation

In this subsection, we only consider 8×8 DCT blocks located in the foreground region. Our objective is to extract the edge strength and orientation information. This can be done by examining the definition of the discrete cosine transform:

$$F_{uv} = \frac{C_u C_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i,j). \tag{4.4}$$

It is easy to verify that the first few AC coefficients represent some specific ways to sum up the pixels in an 8×8 block as shown in Fig. 4.3. Let us take F_{10} as an example. The coefficient is obtained by a weighted summation of 64 pixels, where the top 4 rows take the positive sign while the bottom 4 rows take the negative sign. If F_{10} has a large magnitude, it implies that there is a good chance that we will have a horizontal edge. Similarly, if F_{20} has a large magnitude, then we may have two horizontal edges. The same argument applies to other low frequency AC coefficients as indicated in Fig. 4.3. Shen et al. [46] proposed a rough way to estimate the edge orientation in a block as

$$\tan \theta = \sum_{v=1}^{7} F_{0v} / \sum_{u=1}^{7} F_{u0}. \tag{4.5}$$

This formula is reasonable since the numerator and the denominator indicate the strengths of the horizontal and vertical edges, respectively. Although the obtained edge orientation information is kind of rough, the coarse edge detection technique in the DCT domain was

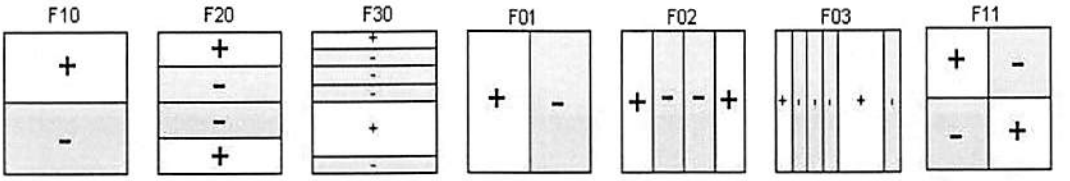


Figure 4.3: The sign patterns of weighted pixel values for the first few AC values.

proved to be about 20 times faster than traditional edge detectors in the pixel domain. Furthermore, the edge strength can be computed according to the following formula:

$$h_{vertical} = \frac{F_{10} + \alpha F_{20}}{9.111 \tan\theta} \quad \text{and} \quad h_{horizontal} = \frac{F_{01} + \alpha F_{02}}{9.111 \tan\theta}. \tag{4.6}$$

Here, we exploit the fast edge orientation estimation algorithm with some modification. That is, we only consider edges that are aligned with the straight lines without any offset from the center of each 8×8 block as shown in Fig. 4.4(a). Furthermore, since it is a block-based estimation, it is difficult to represent a large number of possible edge orientations with good accuracy. Thus, we restrict the estimated edge orientation to eight quantization levels as shown in Fig. 4.4(b). To summarize, we use Eq. 4.5 to compute the edge orientation in blocks of the foreground and then quantize the edge orientation into 8 levels for further processing.

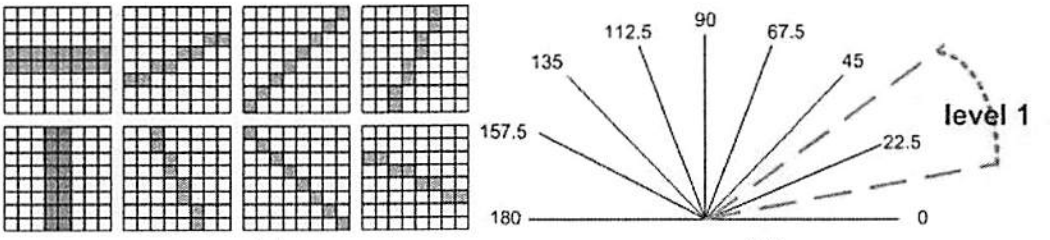


Figure 4.4: The eight quantized levels for coarse-scale edge orientation estimation.

4.1.3 Displacement Parameter Estimation

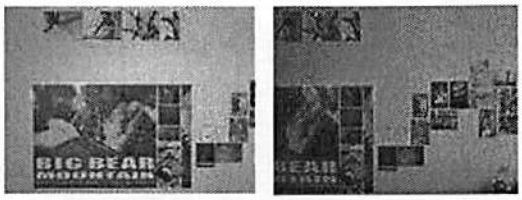
Based on the results obtained from Section 4.1.2, we apply the edge-based image registration technique to determine the displacement parameters to align the left and right two images. One simple way to achieve the alignment is to compute the cross-correlation between the two edge maps, where the edge strength is set to be zero in the background region.

$$r(i,j) = \sum_{j_1}^{j_1<(N_j/2)} \sum_{i_1}^{i_1<(N_i/2)} \left(\tilde{E}_1\left(\left(i_1+\frac{N_i}{2}\right),\left(j_1+\frac{N_j}{2}\right)\right)-m_{\tilde{E}_1}\right) * \left(\tilde{E}_2((i+i_1),(j+j_1))-m_{\tilde{E}_2}\right) \quad (4.7)$$

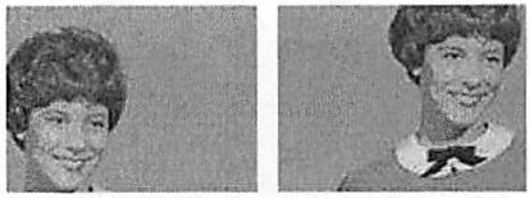
where N_i and N_j represent the width and the height of the overlapping region and $m_{\tilde{E}_1}$ and $m_{\tilde{E}_2}$ are the mean values of the overlapping regions of $\tilde{E}_1$ and $\tilde{E}_2$, respectively. The vector, $r(i,j)$, that leads to the maximal correlation value, gives the optimal displacement in the horizontal and the vertical directions. Since the horizontal or the vertical size of the edge map is 1/8 of that of the original one, the actual amount of displacement should be scaled up by a factor of 8 with respect to the coordinates of the original input images.

4.1.4 Experimental Results

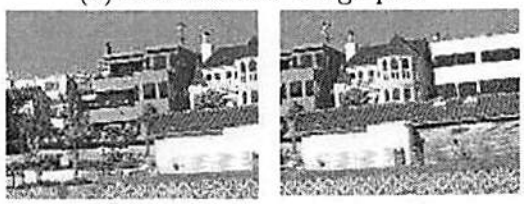
In this section, we present some experimental results with four test image pairs as shown in Figs. 4.5, where (a) and (b) are indoor scenes and (c) and (d) are outdoor scenes (600x448). Generally speaking, images of the outdoor scene usually consist of a higher noise level so that it is more difficult to extract the accurate edge information. The image registration results are usually poorer.



(a)The first test image pair



(b)The second test image pair



(c)The third test image pair

(d)The forth test image pair

Figure 4.5: The four test image pairs.

4.1.4.1 Performance Comparison in Processing Time

Here we compare the two image registration systems using the traditional pixel domain approach and the proposed DCT domain approach. The computational saving comes from two different parts. First, the DCT domain approach avoids the inverse DCT and the forward DCT processes required by the space domain approach. Second, the resolution of the space domain image pair is much finer than that of the DCT domain image pair, i.e. 64 versus 1. Thus, the search for the displacement vector demands much more time. The execution time for the alignment of the 4 test image pairs is compared in Table 4.1 and Fig. 4.6. We see the time saving ranges from 95-97% as compared with the traditional space domain approach.

Table 4.1: Comparison between the proposed and the traditional one in processing time.

	1st	2nd	3rd	4th
traditional	34.2340	36.1410	35.8910	34.1100
proposed	1.0310	1.7030	1.7500	1.6880
save(sec)	33.2030	34.4380	34.1410	32.4220
save(%)	96.9884	95.2879	95.1241	95.0513

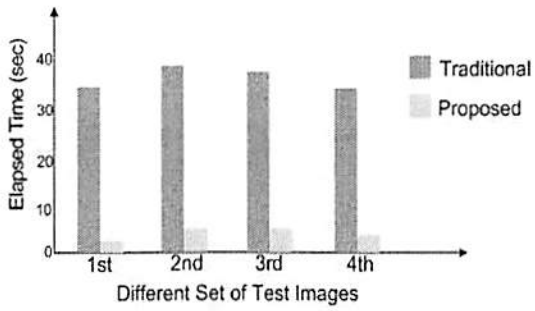
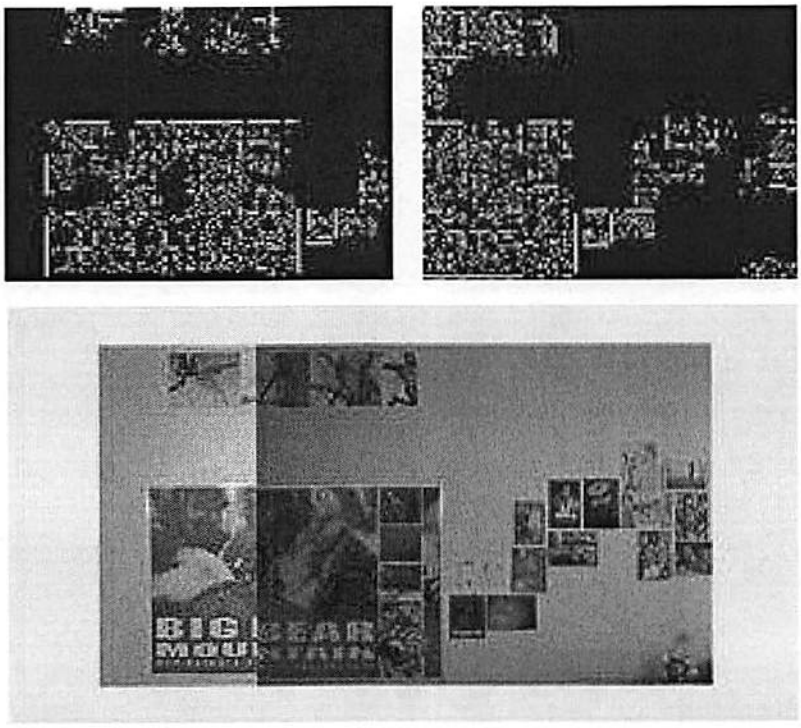


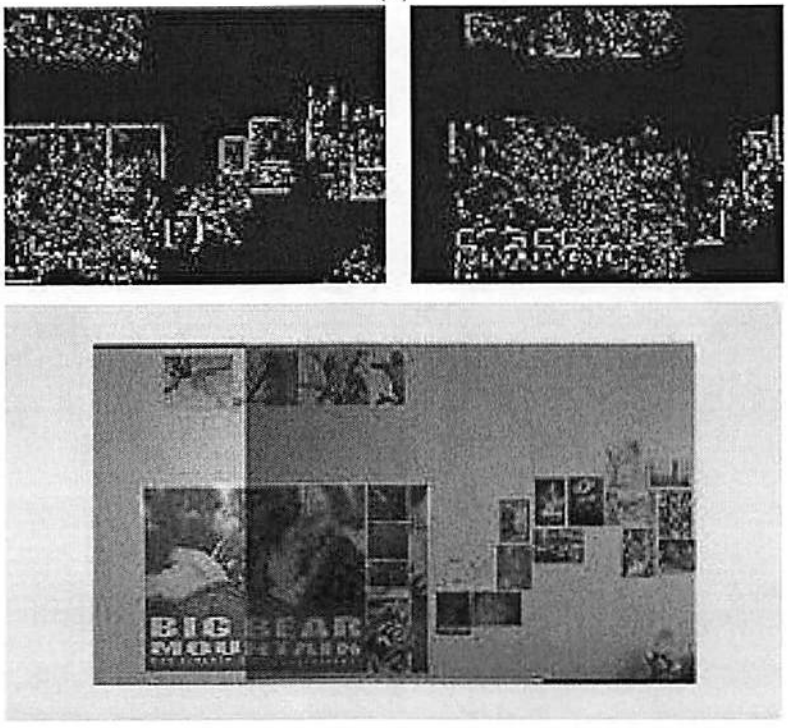
Figure 4.6: Performance comparison in processing time.

4.1.4.2 Comparison of Output Image Quality

We compare the registered output image results in Figs. 4.7-4.10, where (a) shows the results based on the proposed DCT-domain technique and (b) gives the results based on Canny's edge detection. As shown in Fig. 4.7(a), although the estimated edge maps are not as accurate as the ones obtained using Canny's edge detector, the proposed method does catch the general trend well. Besides, the stitched output image in Fig. 4.7(a) looks very similar to the corresponding one in Fig. 4.7(b). Please note that the two input image pairs as shown in Fig. 4.5(a) have a color-mismatch problem. However, the color mismatch does not affect the registration results since the proposed approach does not rely much on the color information. Instead, it is based on the extracted edge with the luminance component only. As shown in Fig. 4.8, the proposed DCT domain approach misses a lot of edge in the facial and cloth areas. Only the edge information in the hair region is caught. This is partly due to the poor performance of the image segmentation step for the foreground extraction and partly due to the limitation of edge detection in the DCT domain. However, since both left and right input images are processed with the same technique, the missed edges do not hurt the image alignment job at the later stage. Actually, the hair information is sufficient to do the alignment. As a result, the proposed



(a)



(b)

Figure 4.7: Image registration results of (a)the proposed DCT-domain and (b)the space-domain approaches.

method can provide almost the same performance as the traditional space domain approach at a much lower computational complexity.

Fig. 4.9 provides the edge detection and image registration results of an image pair of the outdoor scene. The edge maps appear to be quite complicated. However, the proposed technique still provides a simpler edge map to make the later alignment task easier. Furthermore, the two output image mosaics are indistinguishable by human eyes. Finally, for the image pair shown in Fig. 4.5(d), we show the edge detection and the image registration results in Fig. 4.10. By examining Fig. 4.10(a) carefully, we can find that the output image is not perfectly registered in terms of the DCT blocks using the proposed method. We see from the windows of the white building that there exists a misalignment of the two images and the amount of misalignment is around 3 DCT blocks (or 24 pixels). This could be explained by the periodic pattern of roofs that occupy a quite large area of the two input images. Thus, there exist multiple local maxima that make the global maximum selection difficult. One way to fix this problem is to consider multiple thresholds so that we can weigh edges in the window area more to avoid the confusion caused by edges of the roof region. Since the proposed approach is block-based so that a single error in the edge map will lead to a block error in the image domain.

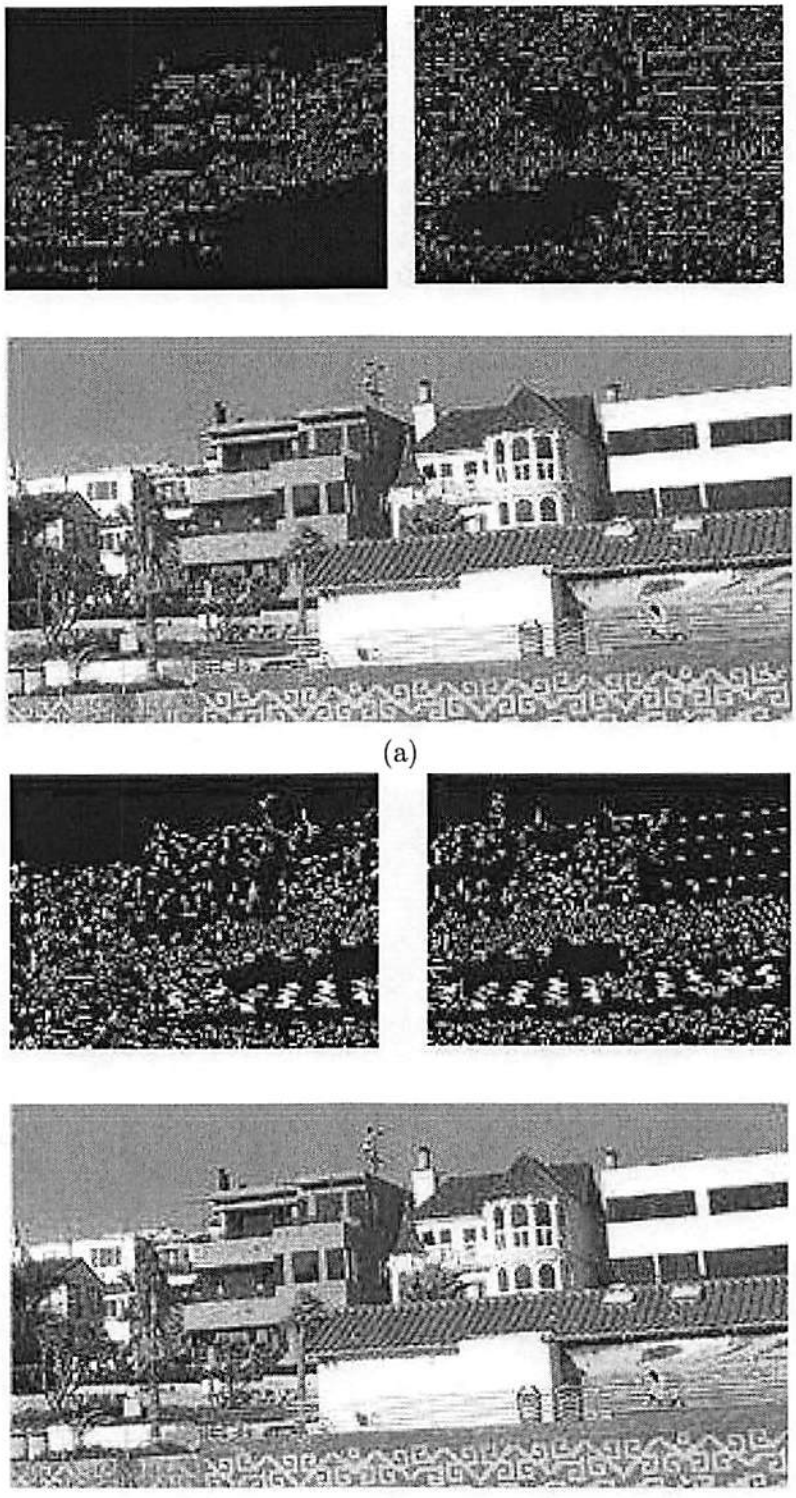
4.2 Block-level Image Registration based on Edge Extraction

A multi-scale DCT-domain image registration technique for two MPEG video inputs is proposed in this section. Several edge detectors are first applied to the luminance component of DC coefficients to generate the so-called difference maps for each input image.

(a)

(b)

Figure 4.8: Image registration results of (a)the proposed DCT-domain and (b)the space-domain approaches.



(a)

(b)

Figure 4.9: Image registration results of (a)the proposed DCT-domain and (b)the space-domain approaches.

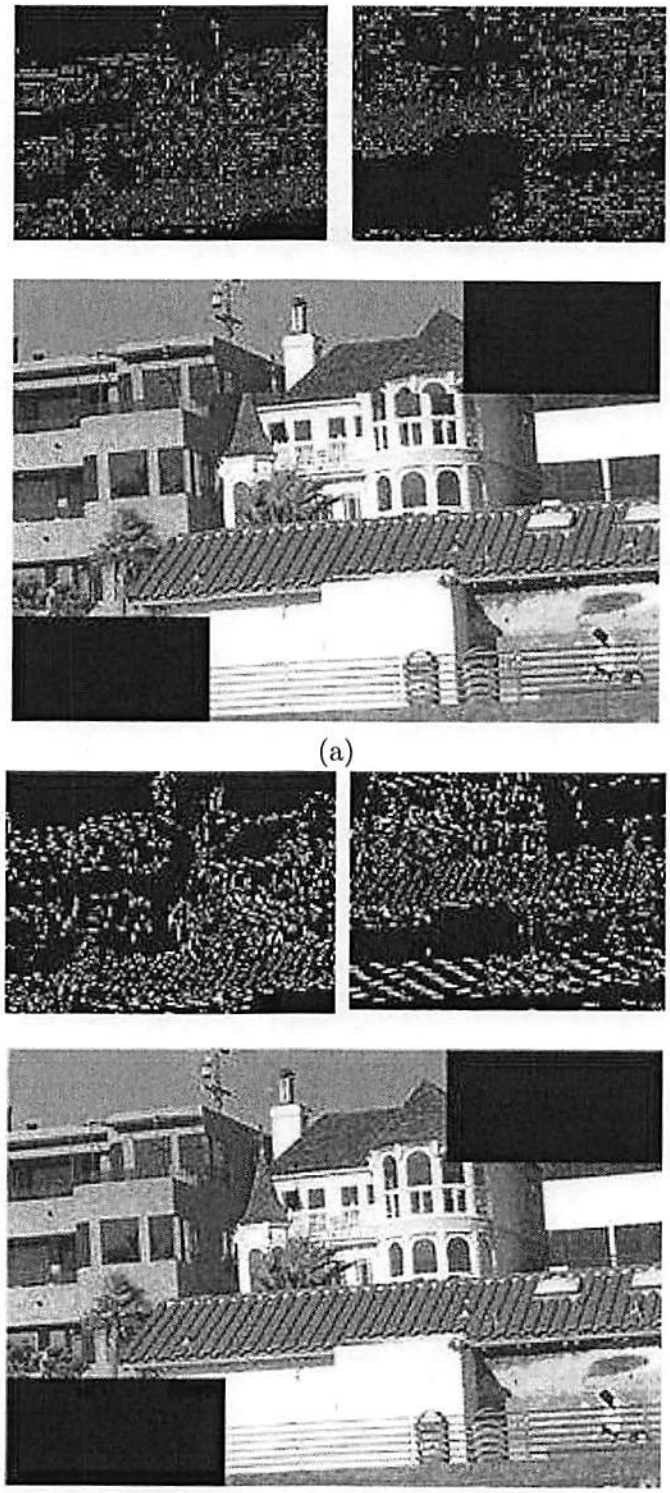


Figure 4.10: Image registration results of (a)the proposed DCT-domain and (b)the space-domain approaches.

Then, a threshold is selected for each difference map to filter out regions of lower activity. Following that, we estimate the displacement parameters by examining the difference maps of the two input images associated with the same edge detector. Finally, the ultimate displacement vector is calculated by averaging the parameters from all detectors. It is shown that the proposed method reduces the computation complexity dramatically as compared to pixel-based image registration techniques while reaching a satisfactory result in composition. The major four parts of the algorithm are detailed in the following sections.

4.2.1 Edge Detection on DC Maps

A DC map of each input image that contains DC values of the luminance (Y) component of all blocks is formed. Since only the DC value is considered for each 8×8 block, the size of the DC map is 1/64 of that of the original image. This means the data we are dealing with are much less than that in the traditional pixel-domain approach. Based on those DC maps, four different edge detectors (H_1, H_2, H_3 and H_4) are applied to each of them. Those edge detectors are:

$$H_1 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

They measure the variation of the image in vertical, horizontal, 45 degree, and 135 degree directions, respectively. Each detector can produce one difference map so that there are four difference maps of each input image. We use D_{ij} to denote the difference map of image $i = 1,2$ with edge detector H_j, $j = 1,2,3,4$. Difference maps are normalized so that all of their values fall between 0 and 1 for further processing. Note that H_1, H_2, H_3 and H_4 are the second-order derivative filters. The first-order derivative filter and the second-order derivative filters in the horizontal direction are given below:

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}.$$

The reason to adopt the second-order derivative filter than the first-order derivative filter

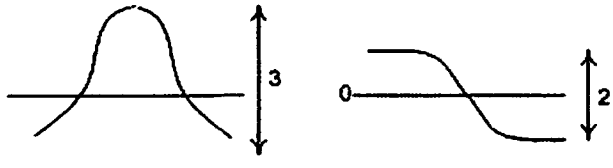
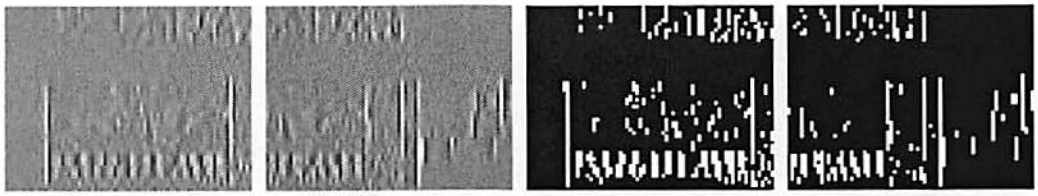


Figure 4.11: Comparison between the first- and the second-order derivative filters.

can be explained using Fig. 4.11. As we can see, the line detector (i.e. the 2nd-order detector) is able to extract more features than the gradient detector (i.e. the 1st-order detector). Note that the detectors are applied to the DC map, we should choose the one that can extract out the region of interest with more active features. Thus the difference maps generated by the 2nd order derivative filters are more suitable for the alignment task in the next stage.



(a) (b) (c) (d)

Figure 4.12: The difference maps of (a)image 1 and (b)image 2 using filter H_1 and the corresponding binary activity maps of (c)image 1 and (d) image 2.

4.2.2 Thresholding

In this step, a content adaptive threshold is set up for each pair of difference maps to generate corresponding binary maps. The main purpose of this step is to filter out minor changes. It reduces confusion and speeds up the following alignment step. The difference maps obtained using filter H_1 for two original DC maps are shown in Figs. 4.12 (a) and (b) while their corresponding binary activity maps, D_{11} and D_{21}, are shown in Figs. 4.12 (c) and (d), respectively. As we see from D_{11} and D_{21}, only the vertical difference are preserved for displacement parameters estimation. Similarly, horizontal, 45 degree and 135 degree features are extracted after applying H_2, H_3 and H_4, respectively. Those features help determine the displacement parameters more accurately and reduce the processing time since the unnecessary detail information has been eliminated.

4.2.3 Displacement Parameter Estimation

Let $P_i \times P_j$ and $N_i \times N_j$ be the sizes of two original input images and their DC maps, respectively, as shown in Fig. 4.2. Then, we have

$$N_i = \frac{P_i}{8} \text{ and } N_j = \frac{P_j}{8}. \tag{4.8}$$

Based on obtained D_{ij}, $i = 1, 2$ and $j = 1, 2, 3, 4$, our task is to determine the alignment parameter for the four sets of binary images. The two-dimensional normalized cross-correlation is computed and the optimal displacement parameter is determined at the position where the maximum value occurs in both vertical and horizontal directions. Let (d_{i1}, d_{j1}) be the parameter pair we get from the binary images obtained by applying detector H_1. Similarly, we have (d_{i2}, d_{j2}), (d_{i3}, d_{j3}) and (d_{i4}, d_{j4}) by following the same procedures. Once those four sets parameters are available, the final estimated displacement can be acquired by either averaging or simply chosing the best one from these four vectors. Then, a coordinate conversion, scaled up by a factor of 8, is performed due to the size difference between the original and binary images. The first three steps of the proposed procedure can be described in a flow chart as shown in Fig. 4.13.

4.2.4 Experimental Results

Experimental results with six test image pairs are shown in this section. Those test images are shown in Fig. 4.14 where (a) and (b) are indoor scenes while (c) to (f) are outdoor scenes with different content complexity, different amount and different types of displacement but all with the same size (600×448). Note that the experimental results show that color mismatch does not affect the quality of registration since the proposed method is not color dependent. Thus, all test images presented here are under the same light conditions to reveal the exact quality of output composition.

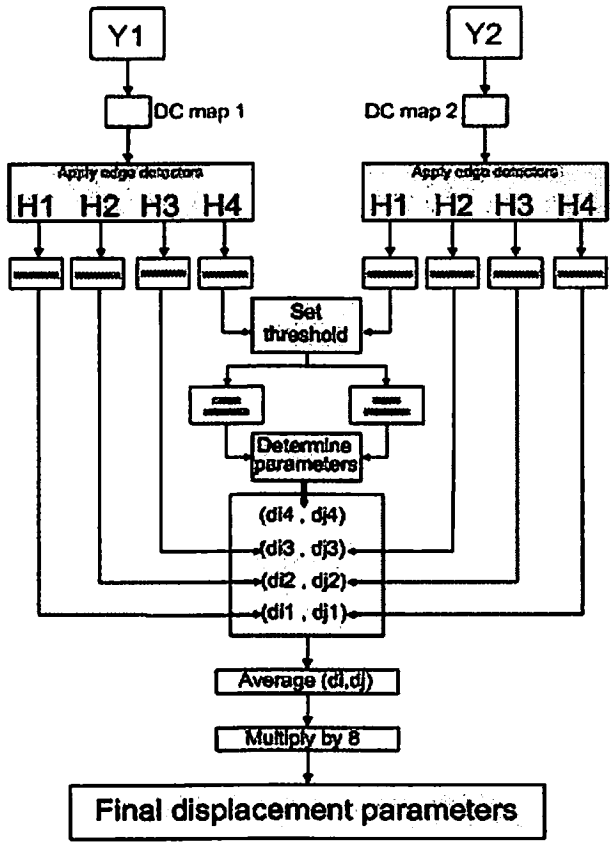


Figure 4.13: A detailed overview of the proposed method.

4.2.4.1 Performance Comparison in Processing Time

The comparison of execution time of the traditional pixel-domain process and the proposed DCT-domain technique is shown in Table 4.2 and Fig. 4.15. The major computation saving of the proposed method comes from two parts: the pixel-DCT domain conversion and information reduction. For the DCT-based method, the time consuming steps, such as inverse DCT and forward DCT, are avoided. Also, as mentioned before, the data being manipulated in the DCT domain has been cut down to 1/64 of the original images so that much less time is required for displacement searching. Those two reasons reduces the processing time over 95% as compared to the traditional pixel-domain approach.

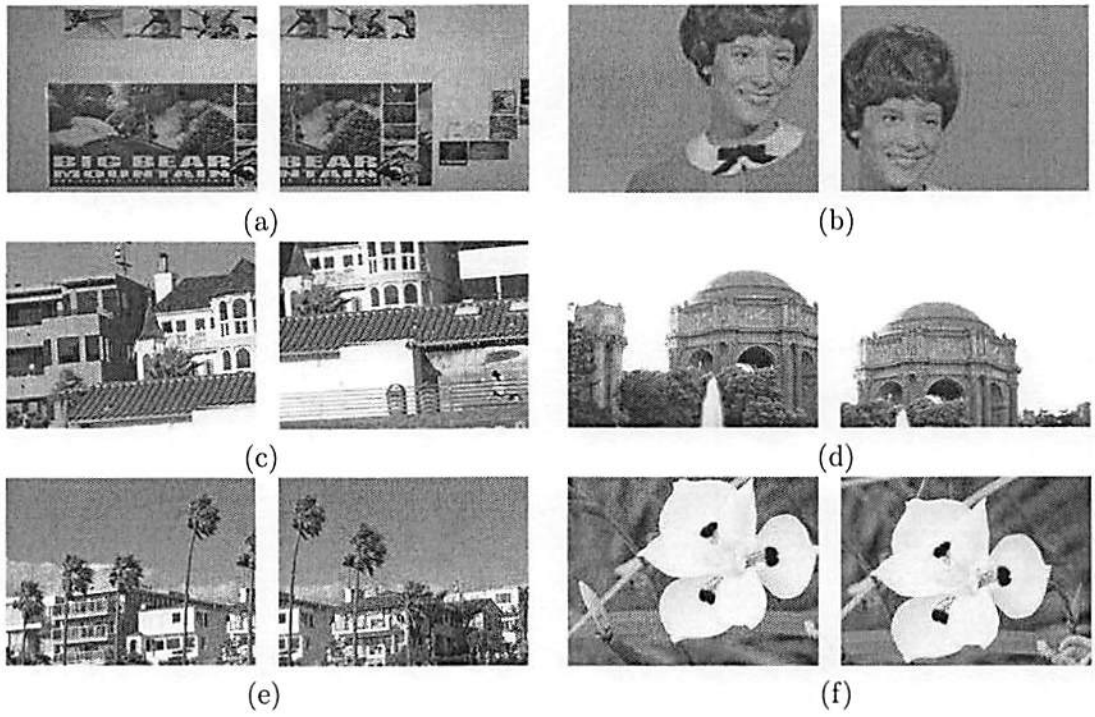


Figure 4.14: The original test images.

Table 4.2: Comparison between the proposed and the traditional approaches in processing time (sec). (a) traditional (b)proposed method. (c) and (d) are the savings in terms of seconds and percentages.

	1	2	3	4	5	6
(a)	35.22	35.28	39.13	35.45	36.11	37.64
(b)	1.407	1.406	1.437	1.516	1.407	1.422
(c)	34.19	34.25	38.13	34.43	35.08	36.58
(d)	97.07	97.07	97.44	97.13	97.14	97.18

4.2.4.2 Comparison of Output Image Quality

The final estimated displacement parameters and composite outputs are shown in Table 4.3 and Fig. 4.16, respectively. Since we know the exact amount of displacement in advance, the estimation errors can be calculated by subtracting the actual ones and the ones that determined by the proposed method. As we see from Table 4.3, estimation errors are within two pixels as compared to the actual displacements. In other words, the precision

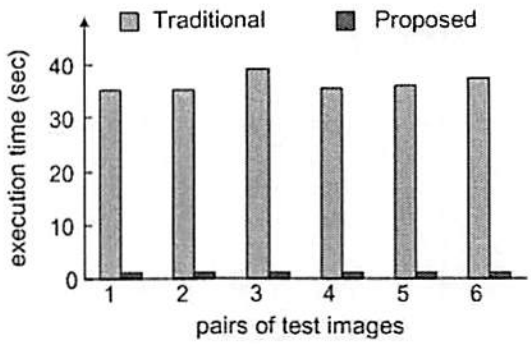


Figure 4.15: Performance comparison in processing time.

can be reached to the sub-block accuracy. The reason why this accuracy can be attained is to consider several displacement parameters generated by different detectors. Each of detectors has a specific directional feature so that the bias among those displacement parameters can be compensated by averaging all of them. In other words, if more detectors applied to the images, more robust parameters we would get. Thus, the pixel accuracy or even the sub-pixel accuracy is possible once appropriate detectors with a good feature can be designed. However, more processing time would be required for applying more filters to the images. We have to find a balance between the processing time and the alignment accuracy.

4.2.4.3 Discussion

Theoretically speaking, the proportion of overlapping area to the original size would affect the quality of the composition since a larger area provides more information such as corners, lines and some other useful features while small area does not. Our experimental results show that, for the same content of images, a larger overlapping area results in more accurate alignment. However, it also depends on how much useful features are within the overlapping parts. If there are only few feature points in the original images, then

(a)

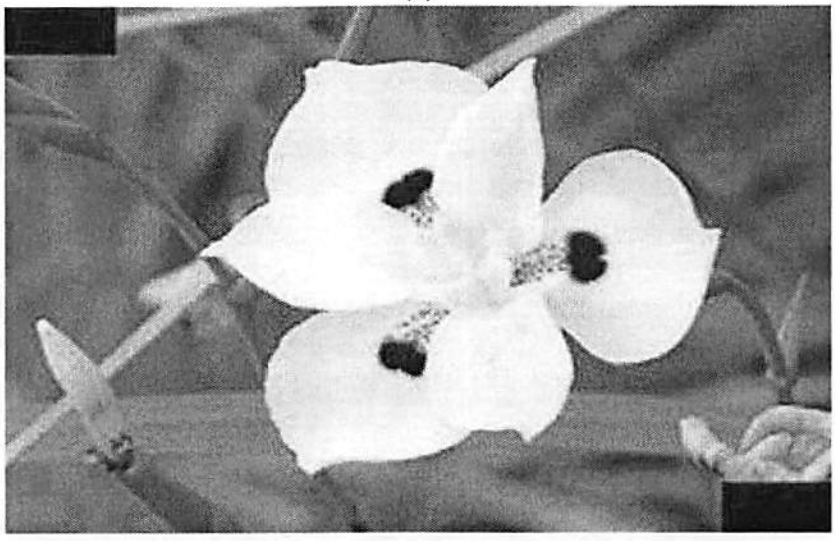
(b)

(c)

Figure 4.16: The stitched images.

(d)

(e)



(f)

Figure 4.17: Figure 4.16 continued.

Table 4.3: Comparison between the displacement parameters (d_i, d_j) derived based on the proposed approach and the actual displacement parameters, (d_i', d_j').

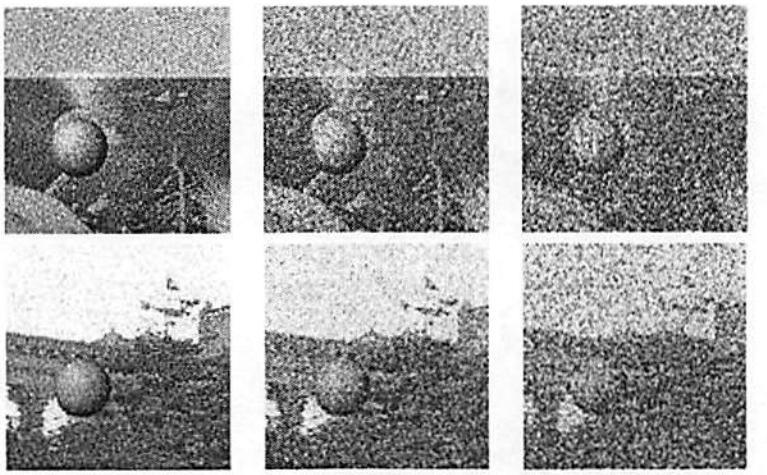
	1	2	3	4	5	6
$8 * d_{j1}$	296	304	304	200	400	504
$8 * d_{i1}$	448	600	304	448	520	496
$8 * d_{j2}$	296	304	304	200	408	504
$8 * d_{i2}$	448	600	296	448	520	496
$8 * d_{j3}$	304	296	296	200	400	504
$8 * d_{i3}$	448	600	296	448	528	496
$8 * d_{j4}$	296	296	296	200	400	504
$8 * d_{i4}$	448	600	296	448	528	496
$8 * d_j$	298	300	300	200	402	504
$8 * d_i$	448	600	298	448	524	496
dj_{actual}	300	300	300	200	400	503
di_{actual}	448	600	300	448	524	495
$8 * (d_j - d_j')$	-2	0	0	0	+2	+1
$8 * (d_i - d_i')$	0	0	-2	0	0	+1

no matter how large the overlapping area is, the performance is similar since the number of useful features is the same. Also note that whether the overlapping area is a multiple of eight affects the quality of composition since if those DCT block are not well aligned, the corresponding DC values of two images at the same position represent different information. In this case, a suitable process, such as interpolation, must be performed in advance.

4.3 Robustness of the Proposed Alignment Method

In this section, the robustness of the proposed alignment algorithm are examined by taking the images with noise as the inputs to the system. The input images of size 480 × 640 considered here are of three different levels of Gaussian noise which are 12.5%, 25% and

37.5%. The DCT-domain registration technique is applied on those images and the experimental results of noise levels being equal to 12.5%, 25% and 37.5% are shown in Fig. 4.18(a), (b), and (c), respectively. Note that the results shown here are the part of the output images that are locally enlarged in order to make it easy to see the quality of the alignment.



(a) (b) (c)

Figure 4.18: The composition results with different levels of Gaussian noise: 12.5%, 25%, and 37.5%.

As revealed in Fig. 4.18, the quality looks good for all cases while the traditional pixel-domain approach can reach high accuracy for the case with noise level up to 12.5%. Our block-based method takes an advantage of dealing with DC maps of the original images. Since the DC map can be treated as a down-sized version of the original image, many features will be averaged out during this process. Therefore, the effect caused by noise can be eliminated and the confusion will be reduced as well while doing the alignment. This verifies the robustness of the proposed block-based alignment algorithm while noise is involved.

Chapter 5

Advanced Mosaic Techniques for Coded Video

In this chapter, we describe three advanced topics for coded video mosaicking: hybrid block/pixel registration, block-based video registration and block DCT analysis and classification.

5.1 Hybrid Block/Pixel Registration

It was shown in Chapter 4 that the block-based registration algorithms save lot of computations using either DC or AC coefficients. However, the accuracy of block-based registration is measured only of the resolution of a block, which is of size 8×8. It is desirable to enhance the resolution of the displacement vector to the pixel-level accuracy. To get such a result, some pixel-domain registration can be made after the block-level registration. In other words, the block-level registration can be viewed as a coarse-level alignment while the pixel-level registration yields the fine-level alignment. However, we do not have to perform the inverse transform on all blocks but some selected blocks to save the computation. The reason is simple. There are blocks that correspond to the flat background so that they do

not carry much information. On the other hand, there are blocks that contain valuable spatial domain information such as edges and corners. Thus, we may perform inverse DCT on these blocks and use the spatial domain information to enhance the registration accuracy. In this case, the computational complexity of this hybrid approach will still be significantly lower than that of the traditional pixel-domain approach.

5.1.1 Alignment of Projected Boundary Blocks

To enhance the accuracy of the alignment, one method is to convert the boundary blocks of two overlapping images back to the pixel domain, add two-dimensional pixel values along horizontal or vertical directions followed by a normalization procedure to get one-dimensional data vectors of both images at the same data point. Then, we can perform the 1D alignment for projected lines. This concept is illustrated in Fig. 5.1. The best match would happen at the position where the maximum correlation value occurs.

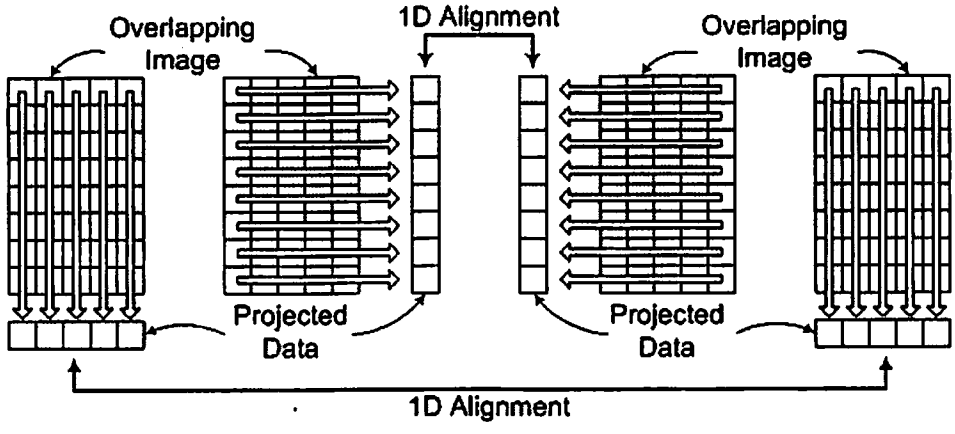


Figure 5.1: Project two-dimensional data to one-dimensional.

It is observed in our experiments that the alignment can be fine-tuned to reach the pixel-level accuracy and the estimation errors are within two pixels. This process is fast

and easy to implement. However, it is not robust in dealing with all kinds of different images since some information may be lost by projecting data from the 2-D domain to the 1-D domain. Also, if the areas of two input images to be transformed back to the pixel domain are not exactly the same, the relevant and irrelevant data could be mixed together to confuse the alignment task. Moreover, if the image content consists of repeated patterns, the normalized data would contain several peaks so that the exact match is more difficult to achieve. The performance of this processing highly depends on the quality of composition obtained from the block-level registration and the image content.

5.1.2 Alignment of Selected 2D Blocks in the Pixel Domain

It is known that salient features of images play an important role in the registration process. Areas without salient features such as the plain background or smooth surfaces contribute little to the final registration result. Thus, we may identify those blocks that contain the salient features in the DCT domain and then transform them back to the pixel domain to fine-tune the coarse alignment result obtained at the block level.

5.1.2.1 Corner Block Detection

As presented in Sec. 4.2, line detectors H_1 and H_2 can filter out simple vertical and horizontal edges. Based on these two types of edge information, corner blocks can be roughly determined by the following procedure.

1. Computing Horizontal and Vertical Edge Maps

   By applying H_1 and H_2 to the DC map of an input image, we get two normlized edge magnitude maps, *i.e.*, D_{H1} and D_{H2}, which takes values between 0 and 1. To

eliminate areas with minor activities such as the background, an adaptive threshold method is applied to the magnitude maps to create binary images B_{H1} and B_{H2}. This threshold value determines the number of blocks of higher activities to be selected in the next step. On one hand, the more blocks selected, the more information provided. On the other hand, the more blocks selected, the higher the computational cost. For example, if higher accuracy is required and a little sacrifice at the complexity is acceptable, the threshold should be set to a lower value. On the contrary, if the computational speed is the main concern, the threshold value should be raised.

2. Computing the Weighted Edge Map

   Given two binary images, B_{H1} and B_{H2}, from the prevous step, we will combine them into a new map using the following weighted scheme:

$$C(i,j) = \omega_1 \times B_{H1}(i,j) + \omega_2 \times B_{H2}(i,j), \quad \omega_1 \neq \omega_2, \tag{5.1}$$

   where $C(i,j)$ has four possible values: 0, ω_1, ω_2 and $\omega_1 + \omega_2$, which mean that the block is a flat block, a block with a strong horizontal edge, a block with a strong vertical edge and a block with strong horizontal and vertical edges, respectively.

3. Decision Making for Corner Blocks

   Once the weighted map C is formed, the next step is to determine which block, $C(i,j)$, has a higher possibility to be a corner block by examining the activities of its eight neighboring blocks. There are several patterns observed that may have one or multiple corners at position (i,j). For $C(i,j) \neq 0$, if its neighboring activities match

one of the patters shown in Fig. 5.2, we claim that this block to be a corner block and set its corner map flag, $B_{corner}(i,j)$, to 1. Otherwise, its corner map flag is set to zero.

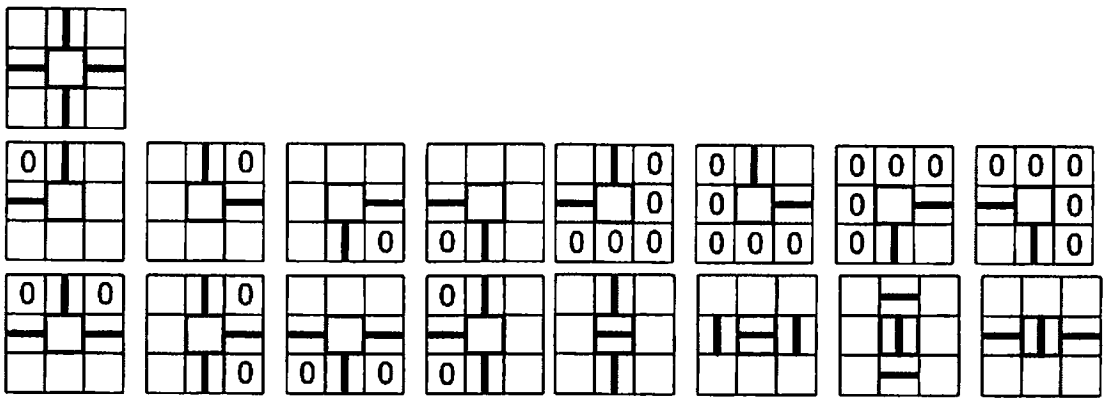


Figure 5.2: The 3 × 3 block patterns that have a higher probability to contain one or multiple corners at the central block.

It is worthwhile to check the validity of the above rule by observing some image examples. To do so, corner blocks in the overlapping region of two input images are checked. First, several corner blocks of one input image are selected automatically using the above rule. Then, their corresponding blocks of the second input image are picked manually. Two cases are examined. The first case is to perform the inverse DCT on selected blocks for examination. The second case is to perform the inverse DCT on those selected blocks as well as their eight neighbors.

In this test example, the two input images are of size 480 × 640, and ten corner blocks are selected from each image. Let f_{i_corner} and f_{j_corner} denote the fine-tuned horizontal and vertical displacement parameters, where $0 \leq | f_{i\_corner} | \leq 7$ and $0 \leq | f_{j\_corner} | \leq 7$, since the coarse scale alignment has been done at the block level. In the example of concern, the coarse-scale displacement vector is equal to (480, 408), which is also the actual

displacement vector. Thus, the fine-scale alignment parameters f_{i_corner} and f_{j_corner} are both equal to zero.

Case 1: Inverse DCT applied to selected corner blocks only

Take one selected corner block of image 1 for instance. After being inverse transformed back to the space domain, two different features, corners and edges, are extracted so that they can be aligned with its corresponding block in image 2.

1. Pixel-wise Matching

   For a selected 8×8 block in the pixel domain, corner detection is performed followed by a pixel-wise comparison. The results are shown in Table 5.1, where each column indicates one alignment result based on each of the 10 selected blocks. As shown in the table, most blocks yields accurate fine-tune parameters. For blocks indexed by 8-10, all 64 pixels in the pixel domain have the same value. That is, the selected block is part of background which provides no clue for fine-scale alignment at all.

Table 5.1: The fine-tuning parameters ($f_{i\_corner}, f_{j\_corner}$) determined by the pixel information of each detected corner block pair.

	1	2	3	4	5	6	7	8	9	10
f_{i_corner}	0	0	0	-7	0	0	-7	N/A	N/A	N/A
f_{j_corner}	0	0	0	-7	0	0	-7	N/A	N/A	N/A

2. Line-wise Matching

   Rather than pixel-wise comparison in a block, the edge information is extracted for the alignment purpose. The alignment results are presented in Table 5.2. Note that there is no "N/A" for all blocks which means more information can be provided by edges.

Table 5.2: The fine-tuning parameters (f_{i_edge}, f_{j_edge}) determined by the edge information of each detected corner block pair.

	1	2	3	4	5	6	7	8	9	10
f_{i_corner}	0	0	0	0.5	0	0	0	0	2	-2
f_{j_corner}	0	0	0	0	0	0	0.75	0	-1.5	-0.5

From the above two tables, we see that most matching pairs can yield the correct results while some cannot. The final displacement parameters can be determined by the best three pairs (rather than averaging the displacement vectors of all 10 sets).

Case 2: Inverse DCT applied to selected corner blocks and their eight neighbors

Intuitively, a larger area should contain more useful information for registration refinement. Thus, a better result is expected in this case.

1. Pixel-wise Matching

   First, the corner information is extracted out for each 24 × 24 area. Then, the fine-tuning parameters are determined by the pixel-wise comparison. The results are given in Table 5.3.

Table 5.3: The fine-tuning parameters ($f_{i\_corner}, f_{j\_corner}$) determined by the pixel information of each detected corner block pairs.

	1	2	3	4	5	6	7	8	9	10
f_{i_corner}	0	0	0	0	0	0	0	0	6	-4
f_{j_corner}	0	0	0	0	0	0	-2	0	2	-2

2. Detecting Edges - Line-wise Matching

The edge information is first extracted in the same area. Then, the fine-tuning parameters are determined by the edge information. The resulting parameters are given in Table 5.4. By comparing results in Table 5.3 and 5.4, we see that the alignment based on the edge information is more accurate since it has smaller errors. The pixel-wise alignment is usually more sensitive to noise. This explains the reason why the line-wise alignment based on the edge information provides a better choice.

Table 5.4: The fine-tuning parameters (f_{i_edge}, f_{j_edge}) determined by the edge information of each detected corner block pairs.

	1	2	3	4	5	6	7	8	9	10
f_{i_edge}	0	0	0	0	0	0.5	0	0	3.75	-1.25
f_{j_edge}	0	0	0	0	0	0	-0.5	0	4.5	-0.5

Since the corner blocks are determined in a downsized image. Sometimes salient features are split between two adjacent blocks. Thus, it is not very reliable to perform the inverse DCT to an isolated block without considering its neighboring blocks. We see from experimental results that the useful information will not be missed if 3×3 blocks centered at the corner block are transformed back to the pixel domain. In summary, the alignment based on the information of a larger area is more robust, and it is better to compare the edge information in these two blocks to determine the fine-scale displacement vector.

5.1.3 Experimental Results

The performance of the hybrid block/pixel registration technique is demonstrated in this section. The execution time comparison of the traditional pixel-domain edge-based method, the proposed DCT-domain alignment technique, and the proposed hybrid block

or pixel alignment method is shown in Table 5.5, where the size of test images 1 to 8 is 448 × 600 and that for test images 9 and 10 is 480 × 640. As compared with the pure DCT-domain alignment method, the hybrid method has to do some extra work, including corner block detection, inverse DCT transform, as well as edge detection in the pixel domain. This is the reason why that the processing time is longer than that of the proposed block-based algorithm. The final displacement parameters are shown in Table 5.6. It is clear that the hybrid block/pixel method improves the accuracy to the pixel level for all test images at the price of increased complexity.

Table 5.5: Execution time comparison (in the unit of seconds) of (a) the traditional method, (b) the proposed DCT-domain algorithm and (c) the proposed hybrid method.

	1	2	3	4	5	6	7	8	9	10
(a)	19.11	19.14	19.53	19.44	19.53	19.45	19.42	20.39	23.02	22.94
(b)	1.36	1.34	1.36	1.34	1.34	1.34	1.36	1.34	1.41	1.36
(c)	9.22	10.61	8.97	9.02	9.44	10.23	9.69	12.03	11.64	12.47

Table 5.6: Comparison of displacement vectors: (d_i, d_j) is obtained by the block-level alignment, $(d_{i\_hybrid}, d_{j\_hybrid})$ is obtained by the hybrid block/pixel alignment and $(d_{i\_actual}, d_{j\_actual})$ is the actual one.

	1	2	3	4	5	6	7	8	9	10
d_i	600	448	448	520	496	448	296	522	480	480
d_j	298	200	296	400	504	306	302	398	408	408
d_{i_hybrid}	600	448	448	524	495	448	296	524	480	480
d_{j_hybrid}	300	200	298	400	503	300	300	400	408	408
d_{i_actual}	600	448	448	524	495	448	296	524	480	480
d_{j_actual}	300	200	298	400	503	300	300	400	408	408
$d_{i\_actual} - d_i$	0	0	0	4	-1	0	0	2	0	0
$d_{j\_actual} - d_j$	2	0	2	0	-1	-6	-2	2	0	0
$d_{j\_actual} - d_{i\_hybrid}$	0	0	0	0	0	0	0	0	0	0
$d_{j\_actual} - d_{j\_hybrid}$	0	0	0	0	0	0	0	0	0	0

5.2 Block-based Video Registration

We assume that inputs to the system are two synchronized MPEG videos at a frame rate of 30 fps. Also, there are only translation differences between them, both containing some moving objects. In order to avoid the ambiguity caused by pure image-to-image alignment and trajectory-based alignment, the input sequences to the proposed system are first segmented into two parts: the static background and the moving objects. Then, they are separately registered based on their associated spatial and the temporal information. The flow chart of the proposed algorithm is given in Fig. 5.3. The unit of the process is 1 GOP (15 fps in our experiments). In other words, the displacement parameters are updated for each GOP. Take the first GOP as an example. After applying four edge detectors (4.2) to the DC map of the I frame, the first set of alignment parameters is determined. Since this is the block-based alignment, a further refinement process is required in order to reach higher accuracy. In the second pass, the motion information of objects from each frame within the same GOP is used to obtain several other sets of refinement parameters. Based on alignment and refinement parameters, we can estimate the final displacement parameter using a weighted average of them.

5.2.1 Static Background Alignment

Based on the I frames of two input sequences, DC maps are available by extracting out the DC coefficients of all blocks in the luminance (Y) component. Since only the DC value is considered for each 8×8 block, the size of the DC map is 1/64 of that of the original image. This means that the data we are dealing with are much less than that in the traditional pixel-domain approach. Based on the information provided by those two DC

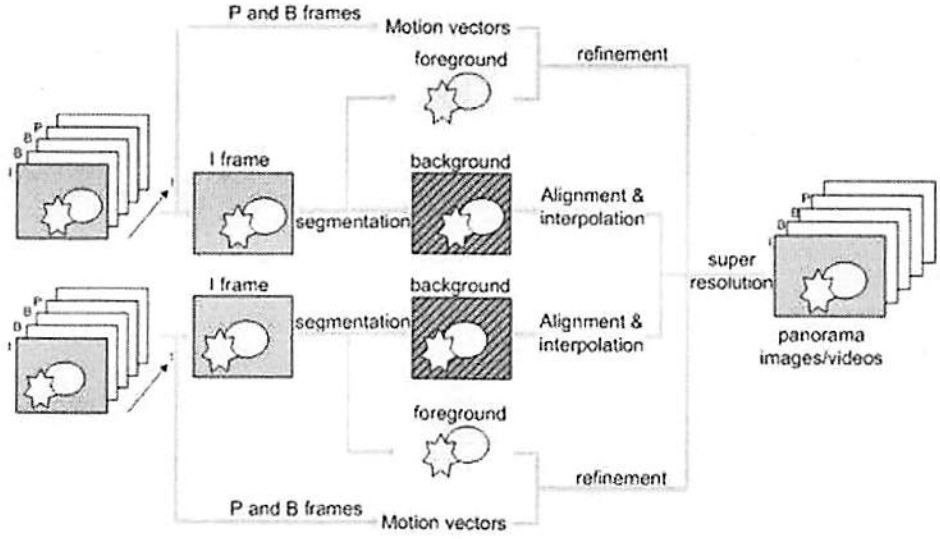


Figure 5.3: The flow chart of the proposed system.

maps, a rough alignment can be done by applying a DCT-domain registration algorithm as described in section 4.2. Four edge detectors are applied on the DC map so that there are four difference maps of each input I frame. For each pair of difference maps, a content adaptive threshold is determined to generate the corresponding binary maps. Based on the four sets of obtained binary images, we determine the alignment parameter by computing two-dimensional normalized cross-correlation followed by a coordinate conversion due to the size difference between the original and binary images. Then, the final estimated alignment parameter, (a_i, a_j), can be acquired by either averaging or simply choosing the best one from these four vectors.

5.2.2 Moving Object Alignment

In this step, the motion vectors of the major moving object obtained from all frames in one GOP are accumulated so that the trajectory can be formed. According to this information,

a trajectory-based alignment process can be applied to enhance the alignment accuracy. Note that the diameter of the bouncing ball in our experiments is around 48 pixels, which corresponds to 3 macroblocks. Thus, in order to avoid incorrect information provided by motion estimation, the motion vector is not considered if its size exceeds a predetermined threshold value (which is set to 3 times the macroblock size in the given example). Once the candidate motion vectors of each frame are determined, a one-dimensional correlation-based sequence alignment process is performed and the optimal parameter is determined at the position where the maximum value occurs. Since the GOP of the input sequence consists of 15 frames, we have 14 refinement parameters for each GOP in total, denoted by (r_{ik}, r_{jk}), $k = 1, 2, \cdots, 14$. There exists a tradeoff between the size of the moving object and the speed of the process. Usually, a larger moving object is preferred since it clearly and strongly represents the behavior of the cluster of macroblocks that contains the object. That is, it is easy to tell whether a macroblock belongs to the actual moving object or just an estimation error. However, in this case, we have to consider more motion vectors, which requires some more processing time. On the other hand, if the moving object is not that big, say within one macroblock, then only one motion vector is taken into consideration. Even though the computational complexity is lower, the robustness of the estimation result is also lower.

5.2.3 Displacement Parameter Estimation

Following the procedures described in the last two subsections, coarse-alignment and motion-based refinement parameters, (a_i, a_j) and (r_{ik}, r_{jk}), $k = 1, 2, 3, \cdots, 14$, are obtained. The final displacement parameter, (d_i, d_j), can be computed as

$$(d_i, d_j) = \alpha \times (a_i, a_j) + (1 - \alpha) \times \left[\frac{1}{14} \times \sum_{k=1}^{14} (r_{ik}, r_{jk}) \right]. \tag{5.2}$$

In words, (d_i, d_j) is a weighted average of (a_i, a_j) and (r_{ik}, r_{jk}), $k = 1, 2, 3, \cdots, 14$. In our experiments, we tried different values of α and found that $\alpha = 0.5$ provides a reasonably good result. The same procedure is applied to all GOPs of the input sequences. Note that the GOP of the generated input videos is 15 frames and since the frame rate is 30 fps (frames/sec), one can update the displacement parameters every I frame, *i.e.* every 0.5 sec. Thus, if there is an error occurring in P and B frames, it will not propagate for too long so that severe visual degradation of the output can be avoided. If there is an abrupt scene change occurring in one GOP, the residual signal in one particular frame will become quite large. It is not difficult to find a threshold to detect such a scene change frame. Then, we are able to split the GOP into two separate parts. Thus, the proposed alignment process can be applied to each individual part separately.

5.2.4 Experimental Results

For the first example, the leading I frames of the two input MPEG2 sequences are shown in Fig. 5.4. As shown in this figure, the moving object is a yellow bouncing ball in front of a poster with a horizontal translation motion only. Fig. 5.5 shows the portion of the

15^{th}, 30^{th}, and 45^{th} stitched frames from the two input sequences. The displacement parameters determined by the I frames and motion vectors of the first three GOP's are $(410, 480)$, $(408, 480)$, and $(410, 480)$, respectively. Thus, we are able to use the background and motion information to do the alignment to generate a mosaic video of high quality.

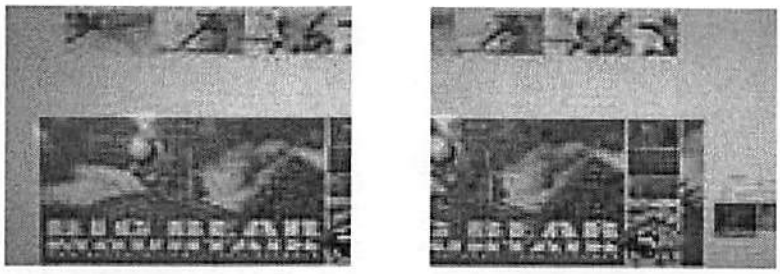


Figure 5.4: The first frames of two input sequences for the 1st experiment.

(a) (b) (c)

Figure 5.5: The portion around the boundaries of stitched frames: (a) the 15th frame, (b) the 30th frame, and (c)the 45th frame.

The two input sequences for the second example are outdoor scene as shown in Fig. 5.6. The 15^{th}, 30^{th}, and 45^{th} stitched frames are shown in Fig. 5.7. We see that these stitched frames have good quality. When comparing obtained displacements with the actual ones, we observe that the estimation errors are no larger than one half block (*i.e.* 4 pixels).

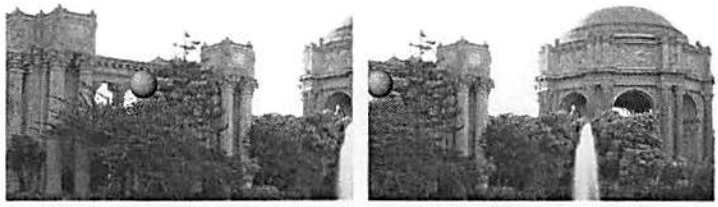


Figure 5.6: The first frames of two input sequences for the 2nd experiment.

(a) (b) (c)

Figure 5.7: The portion around the boundaries of stitched frames: (a) the 15th frame (b) the 30th frame and (c)the 45th frame.

5.2.4.1 Discussion

Generally speaking, the proposed DCT-domain and motion vector-based alignment cannot provide sufficiently accurate information to reach 100% alignment accuracy since they are either block- or macrblock-based features. Some estimation error will result. However, by averaging and weighting, those effects can be reduced to some satisfactory degree. Also, our algorithm belongs to area-based alignment techniques, which is usually more robust than feature-point based alignment since some feature points may disappear in one of two frames and feature tracking is not easy.

Experimental results show that certain accuracy can be reached in the first step based on the alignment of DC coefficients alone in most cases. However, the proportion of the overlapping area to the original size and how many useful features are within the

overlapping parts would affect the quality of the composition. If there are few textured feature points in original images, the performance degrades. On the other hand, if the overlapping region contains highly regular periodic textured patterns, the accuracy of the alignment will decrease, too. In the second step, since only moving objects are considered, the characteristics of the objects plays an important role.

The two steps of the proposed algorithm are both conducted using coded video data. Thus, we do not have to seek additional image/video features and the tedious conversion between the spatial and compressed domain can be avoided. As a result, we can save a lot of computation. Also, since only DC coefficients are taken into consideration for rough alignment, it can be treated as a downsized version of the original image with the factor of 1/64. For those two reasons, the computation complexity is reduced a lot when it is compared to the traditional spatial domain processes. This is the main advantage of the proposed algorithm.

5.3 DCT Block Analysis and Classification

Traditional image registration techniques can be categorized into two groups: feature-based and area-based. Both of them are conducted in the pixel domain. In other words, the detection process is performed over the whole image. To save the computation complexity, we may find an efficient way to analyze the image block content in the DCT domain so that different processing techniques can be applied to blocks of different characteristics. For example, an image can be classified into the background, textures, edges, and so on. Then, based on the group type, we can treat them differently.

5.3.1 DCT-Domain Block Classifications

A block classification scheme based on the DCT domain information is proposed here. By examining the definition of 2D (two-dimensional) DCT transform for an 8×8 block given below

$$F_{uv} = \frac{C_u C_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i,j), \tag{5.3}$$

we see that each DCT coefficient is a linear combination of 64 basis functions as shown in Fig. 5.8.

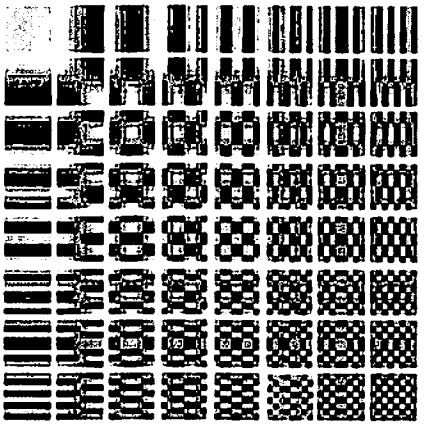


Figure 5.8: The 8×8 array of basis images for the 2D DCT.

Each basis function has different vertical and horizontal space frequencies. The most left upper corner, F_{00} is called the DC coefficients and the rest 63 coefficients, F_{ij} are called AC coefficients. The DC coefficient represents the weighting of the lowest frequency within the 8×8 block while the most right bottom AC coefficients represents the weighting of the highest space frequency. In other words, a guess of the content of this 8×8 block in the pixel domain can be made by observing those 64 DCT coefficients.

As shown in Fig. 5.9, some groups are formed according to some specific geometric properties.

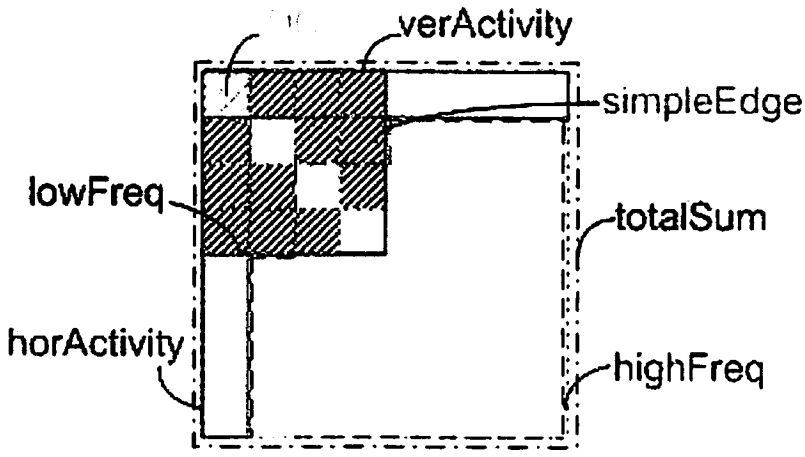


Figure 5.9: Area grouping of DCT coefficients for defining the ratios for block classification.

- F_{ij} *for* $i = 0, \ldots, 3$ *and* $j = 0, \ldots, 3$, are clustered as $G_{lowFreq}$. These shadowed blocks form a group called $G_{simpleEdge}$, which contains only simple vertical and horizontal edges.

- The first row, F_{0j} *for* $j = 0, \ldots, 7$, characterizes the behavior of vertical edges while the first column, F_{i0} *for* $i = 0, \ldots, 7$, illustrate the activity of horizontal edges. They form the groups called $G_{verEdge}$ and $G_{horEdge}$, respectively.

- The remaining blocks are grouped as $G_{highFreq}$ which consists of blocks with high space frequency and the whole 8×8 block is defined as G_{total}.

Let $s_{lowFreq}$, $s_{simpleEdge}$, $s_{horEdge}$, $s_{verEdge}$, $s_{highFreq}$, and s_{total} denote the summations of the total energy of each group. Then, some ratios can be defined for block classification. They are given as

$$\begin{aligned}
R_{DC}(i,j) &= \frac{DC(i,j)}{s_{total}(i,j)}; & R_{simpleEdge}(i,j) &= \frac{s_{simpleEdge}(i,j)}{s_{total}(i,j)} \\
R_{lowFreq}(i,j) &= \frac{s_{lowFreq}(i,j)}{s_{total}(i,j)}; & R_{highFreq}(i,j) &= \frac{s_{highFreq}(i,j)}{s_{total}(i,j)} \\
R_{verEdge}(i,j) &= \frac{s_{verEdge}(i,j)}{s_{total}(i,j)}; & R_{horEdge}(i,j) &= \frac{s_{horEdge}(i,j)}{s_{total}(i,j)} \\
& & for\ 0 \le i \le h \quad & and \quad 0 \le j \le w
\end{aligned} \tag{5.4}$$

where h and w are 1/8 of the height and the width of the original images, respectively, and the values of those ratios are between 0 and 1.

An appropriate threshold is set for each ratio so that the weak activities of each group can be eliminated. For example, suppose that only the top 10% of those edge blocks are needed, a threshold is adopted to filter out the 90% of blocks which have smaller ratio values. These blocks are called inactive blocks and they are not going to be taken into consideration for the following steps so that the computation complexity can be saved to some degree.

After the threshold for each group is defined, every block can be categorized into a different group by following the tree structure as given in Fig. 5.10. Note that the blocks of each group have relative strong strength with respect to a specific geometric property. The block classification diagram can be explained below.

- A block in an image is first separated into the plain background and complex areas according to the R_{DC} value. Since the DC value can be treated as an average of each 8×8 block and its corresponding geometric pattern is a plain area with its space

frequencies equal to (0,0). Thus, if the DC energy dominates the total strength of the block at position (i,j), *i.e.* higher $R_{DC}(i,j)$, then the block has high possibility to be a part of background or smooth area without containing any useful information for registration.

- For blocks in the group of complex areas, they can be further categorized into the texture group and the non-texture group by setting an threshold on $R_{highFreq}$ since the texture blocks usually have higher space frequencies.

- As to the non-texture group, some blocks might contain only simple edges which are purely vertical or horizontal edges and can be extracted out by considering $R_{simpleEdge}$, *i.e.* the behavior of the fist few AC coefficients. If the vertical edges are of interest, then blocks with only vertical edges can be taken out from the group of simple edges.

One advantage of this tree structure is that one can choose 'any leaf' of it. In other words, blocks can be classified based on different features. Each path to the leaf is just a combination of several decisions. For the results of horEdge/verEdge, we can even use the method proposed before to compute the edge strength and classify blocks into even smaller groups.

5.3.2 Experimental Results

The experimental results of two test images of size 480 × 640 are displayed in Fig. 5.11 and Fig. 5.12. In these figures, blocks with intensity one in (a) and (b) show the extracted background areas, blocks containing texture are displayed in (c) and (d), and blocks with

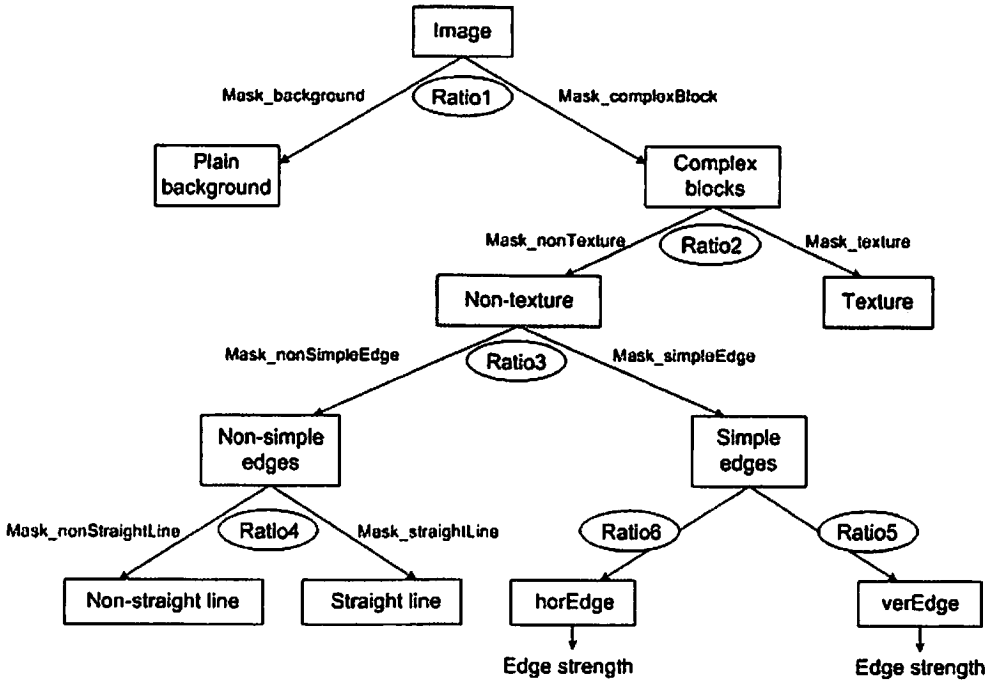
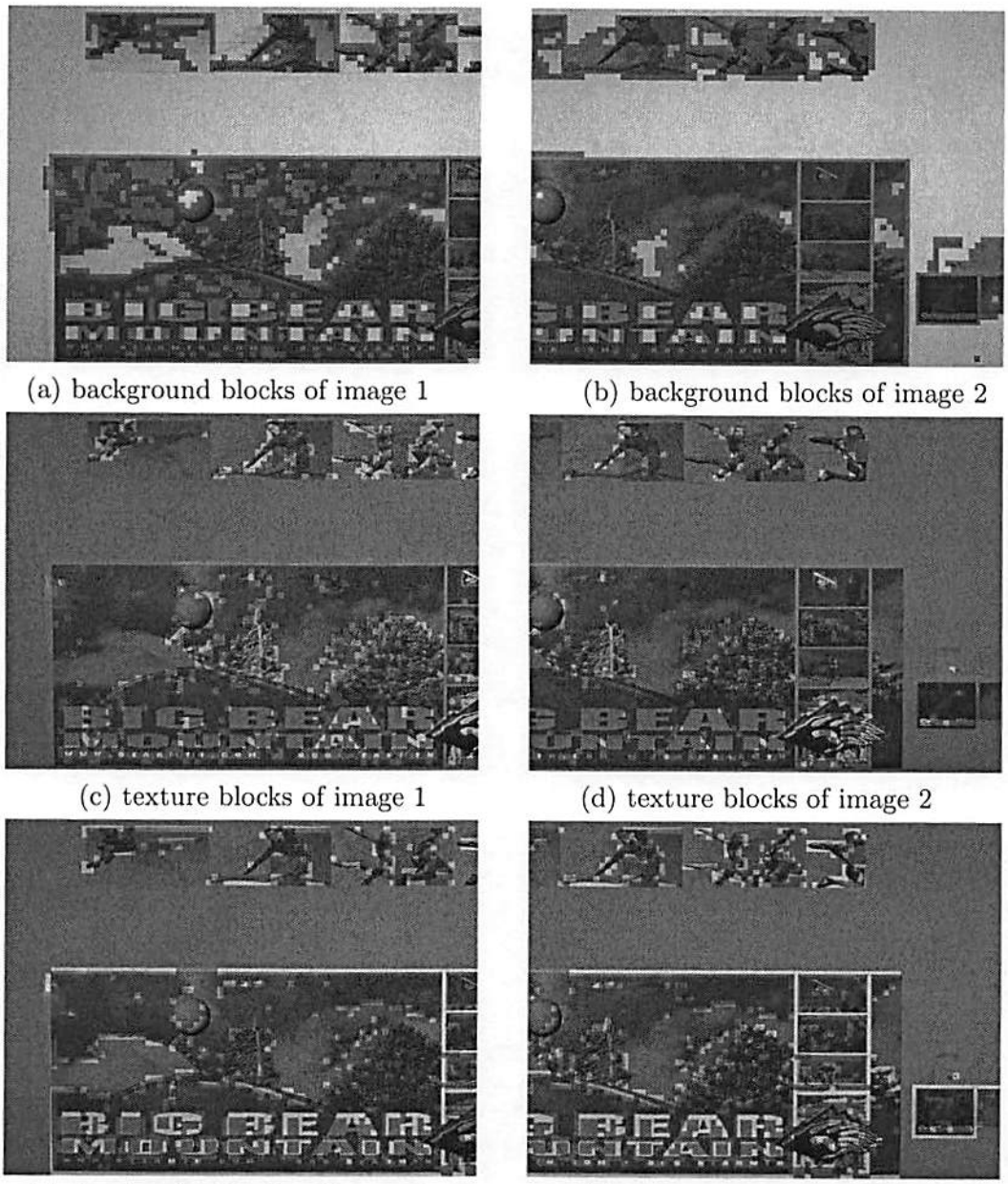


Figure 5.10: The block classification diagram.

simple edges are shown in (e) and (f). The background are perfectly extracted for both cases so that some processing tasks, such as registration and super resolution, can be skipped for those blocks. For blocks in the texture group, they can also be ignored in registration since the repetitive pattern may cause confusion in the alignment process and human eyes are not sensitive to the displacement of the textured region. It is apparent that more attention should be paid to blocks containing edges since they provide useful features for registration. Sharp edges are especially preferred. For reasons described above, different weights can be assigned to blocks of different characteristics.

To conclude, features are first extracted and more complicated enhancement techniques are applied only to those blocks that have a higher weight. Then, the total computational complexity can be reduced on the blocks of less importance while the visual quality remains satisfactory.



(a) background blocks of image 1

(b) background blocks of image 2

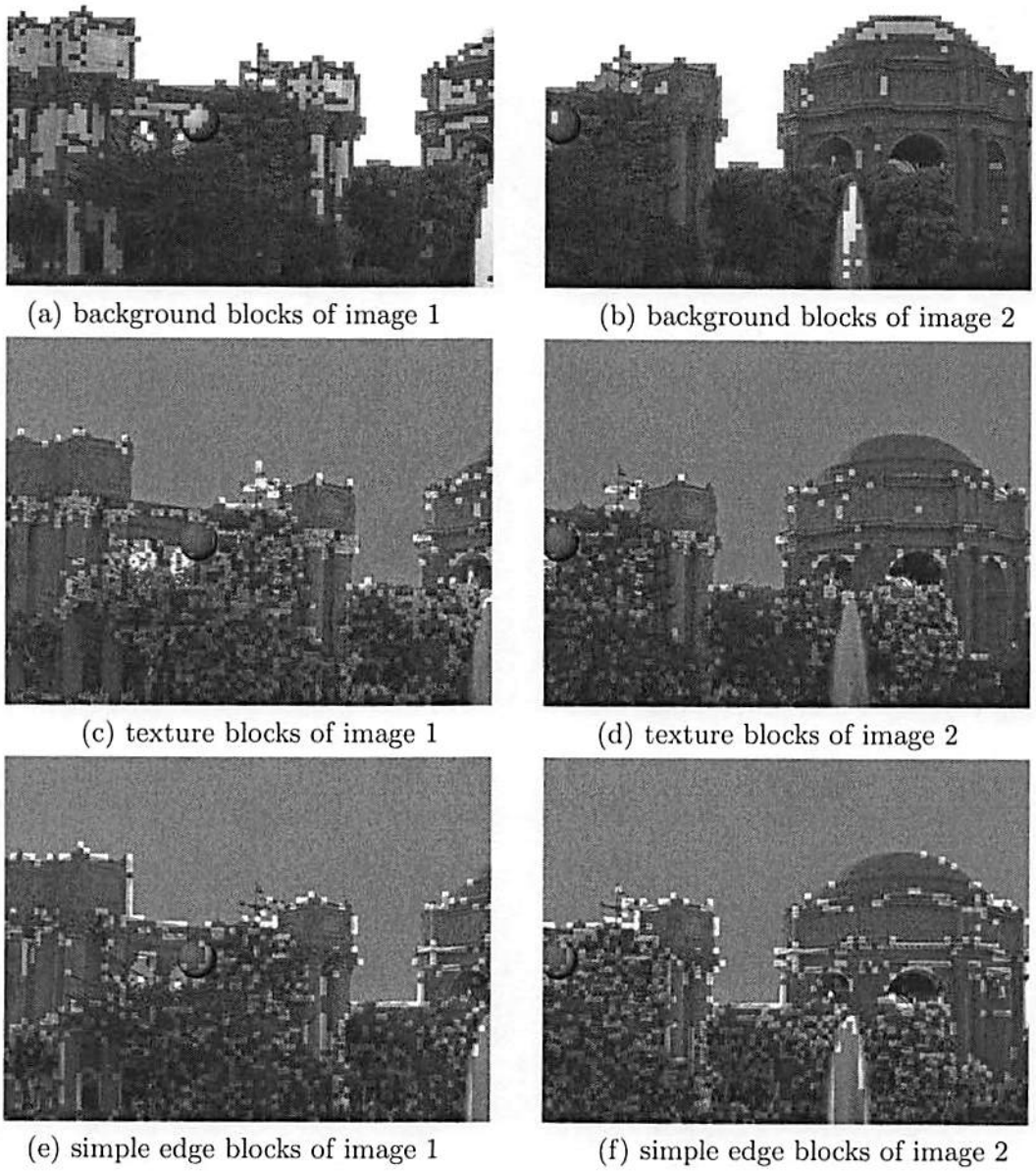
(c) texture blocks of image 1

(d) texture blocks of image 2

(e) simple edge blocks of image 1

(f) simple edge blocks of image 2

Figure 5.11: The block classification results - 1st test image.



(a) background blocks of image 1

(b) background blocks of image 2

(c) texture blocks of image 1

(d) texture blocks of image 2

(e) simple edge blocks of image 1

(f) simple edge blocks of image 2

Figure 5.12: The block classification results - 2nd test image.

Chapter 6

Block-Adaptive Image Upsampling and Enhancement Techniques

Several techniques for color compensation and registration of coded images were introduced in previous chapters. The proposed methods provide efficient ways to correct the color distortion and composite images to form a panorama output. Note that the degradation of the image/video quality during the capturing process has not yet been considered in both cases. If the image/video contents are displayed among various electronic devices of different resolutions, quality degradation may become severe. To overcome this problem and generate an output image of higher quality, super resolution and image enhancement techniques are discussed in this chapter. These techniques will be integrated with the DCT-domain block classification technique to lead to an integrated enhancement system.

Block classification, which can be conducted in either the pixel domain or the DCT domain, categorizes each image block into several types. It serves as a pre-processing step to analyze the image content so that different processing techniques can be selected for different groups. In this chapter, image upsampling and enhancement techniques are

developed based on block classification results. They are explained in Section 6.1 and Section 6.2.

6.1 Block-Adaptive Image Up-Sampling Techniques

A block-adaptive super resolution technique for image up-sampling is proposed in this section. Several issues are examined, including the computation complexity, visual quality and the difference between the original HR image and the image with down- and up-sampling.

6.1.1 Complexity Comparison

With the Maximum A Posteriori (MAP) approach as the backbone for the image upsampling system, we would like to compare the computational complexity between traditional image-based and block-adaptive approaches. The whole image is treated as a single data vector in the image-based method. In contrast, the image is divided into several blocks of equal size in the block-adaptive method, where each sub-block is viewed as a small image for individual processing. Block size 8×8 is chosen here for its compatibility with prevalent image/video coding schemes.

Different interpolation methods, including the zero-order-hold (ZOH), bilinear interpolation (BLI), block-adaptive super resolution (BSR) and traditional MAP estimation (MAP), are applied to several images with different sizes for the processing time comparison in Table 6.1. Note that the original image is treated as a data vector for the zero-order-hold, bilinear interpolation and MAP methods, while the original image is divided into several 8x8 blocks in the block-adaptive algorithm. As shown in Table 6.1, the

zero order hold method and bilinear interpolation methods have a lower computation cost as compared with the block-based super resolution algorithm or traditional MAP method.

Table 6.1: Comparison of processing time (sec.) of different interpolation methods, including the zero-order-hold (ZOH), bilinear interpolation (BLI), block-adaptive super resolution (BSR) and traditional MAP estimation (MAP).

Image Size	*ZOH*	*BLI*	*BSR*	*MAP*
8×8	0.0161	0.0160	0.5620	0.5620
16×16	0.0162	0.0150	2.1250	3.4852
32×32	0.0150	0.0161	6.4220	119.8280
64×64	0.0320	0.0620	23.8910	N/A
128×128	0.2030	0.1250	80.4690	N/A
256×256	1.2650	1.4680	241.2650	N/A

Fig. 6.1 shows the relationship between the image size and the normalized processing time for each interpolation method. Note that the x-axis denotes the original image size that ranges from 8×8 to 256×256 while the y-axis represents the normalized processing time for each method (in the unit of seconds per pixel). As shown in Fig. 6.1, the normalized processing time of the zero order hold or the bilinear interpolation does not fluctuate a lot as the image size increases. For traditional image-based MAP (the black line) as shown in Fig. 6.1, the computational complexity increases dramatically as the input image size increases. For an image of size $N \times N$, it will be represented by an $N^2 \times 1$ vector as the input to the MAP function. Then, its gradient (1st-order derivative) is an $N^2 \times 1$ data vector and the 2nd-order derivative would be of size $N^2 \times N^2$. When the image size N becomes larger enough, the matrix size will be of $O(N^4)$, which explains the lack of experimental data for MAP when the image size is larger than 64×64 in the table and the figure. In the proposed block-adaptive algorithm, we apply different interpolation techniques based on different block types. Since bilinear interpolation has a

lower computation complexity than MAP, the proposed algorithm has a lower complexity than MAP.

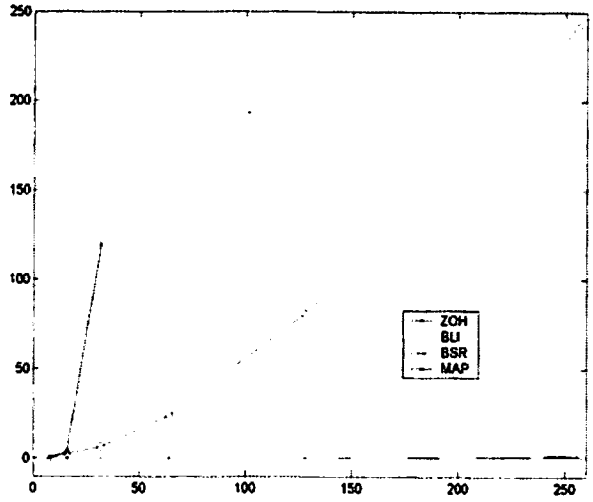
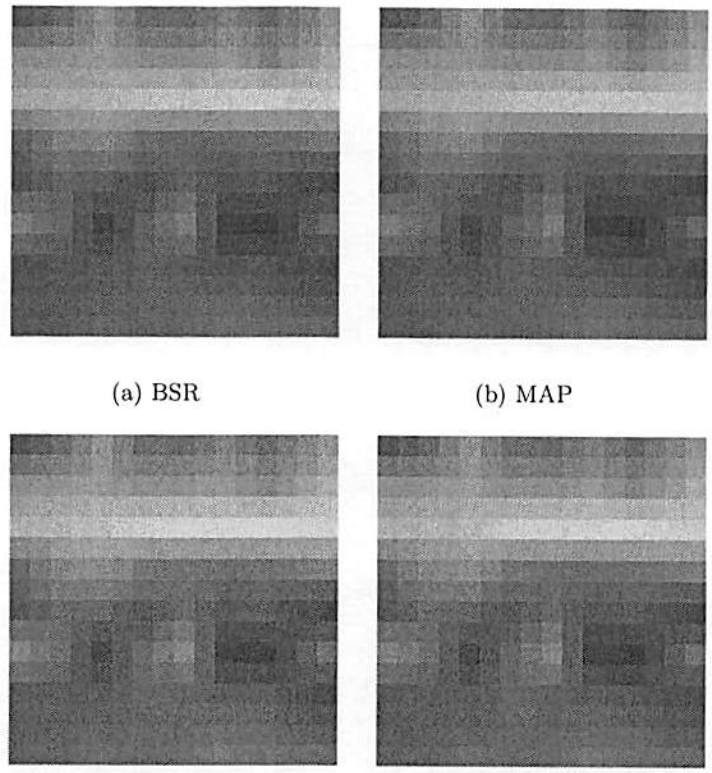


Figure 6.1: The complexity for different methods measured in terms of the processing time (in the unit of seconds) as a function of the image size (in the unit of pixels).

6.1.2 Visual Quality Comparison

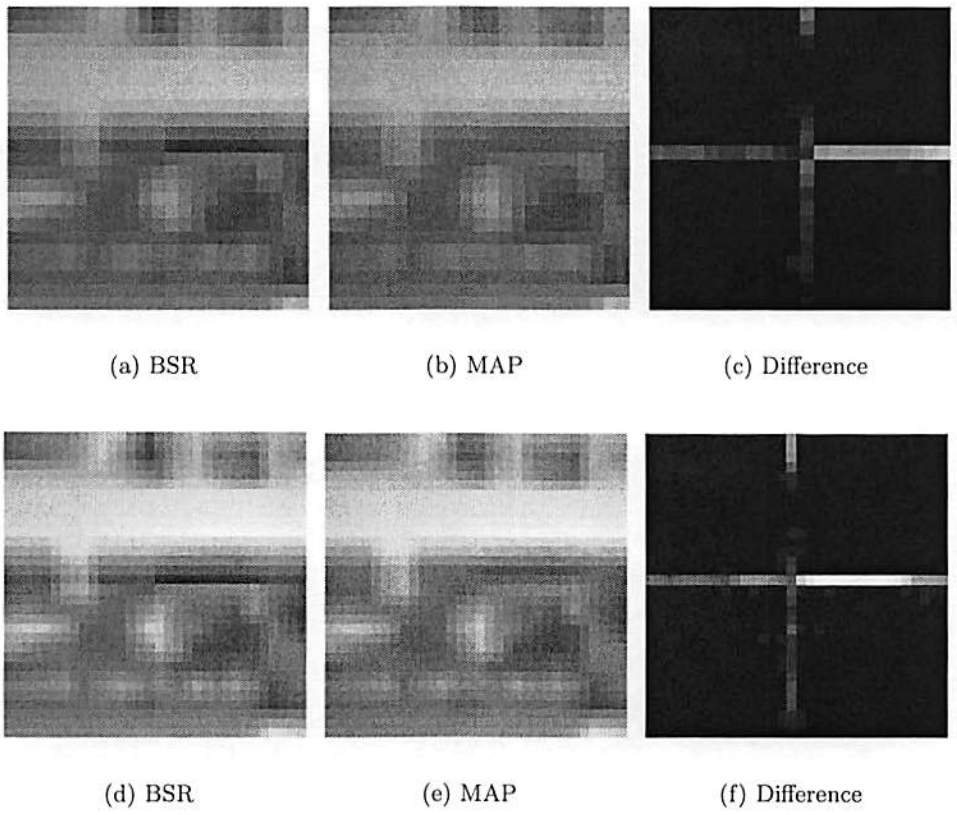
By comparing the block-adaptive algorithm and the traditional MAP method, we see that their outputs have similar perceptual quality except that the formal one has blocking artifacts. As shown in Figs. 6.3 to 6.5, the difference between these two results lies only in boundary areas. Moreover, there is little difference when the algorithm is applied in either the RGB domain or the YCbCr domain. This is because super resolution techniques are more related to the spatial characteristics but less to the color characteristics. Thus, we can choose either the RGB or the YCbCr domain to apply the superresolution techniques.



(a) BSR (b) MAP

(c) BSR (d) MAP

Figure 6.2: Visual quality comparison of different image-upsampling methods for blocks of size 8×8 in RGB domain ((a) and (b)) and in YCbCr domain ((c) and (d)).

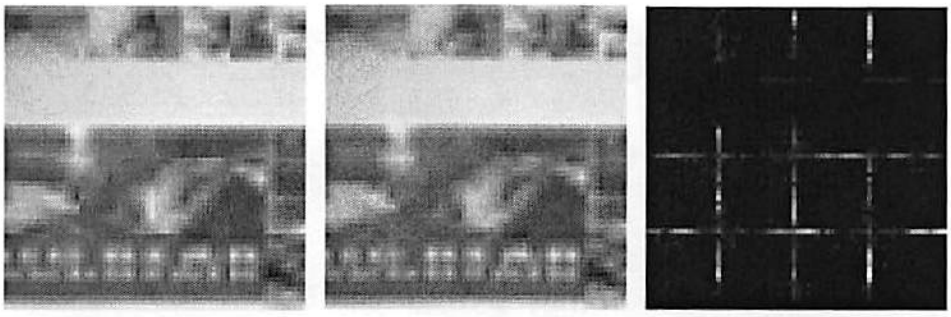


(a) BSR (b) MAP (c) Difference

(d) BSR (e) MAP (f) Difference

Figure 6.3: Visual quality comparison of different image-upsampling methods for blocks of size 16×16 in RGB domain ((a) and (b)) and in YCbCr domain ((d) and (e)). (c) and (f) are difference maps.

(a) BSR (b) MAP (c) Difference



(d) BSR (e) MAP (f) Difference

Figure 6.4: Visual quality comparison of different image-upsampling methods for blocks of size 32×32 in RGB domain ((a) and (b)) and in YCbCr domain ((d) and (e)). (c) and (f) are difference maps.

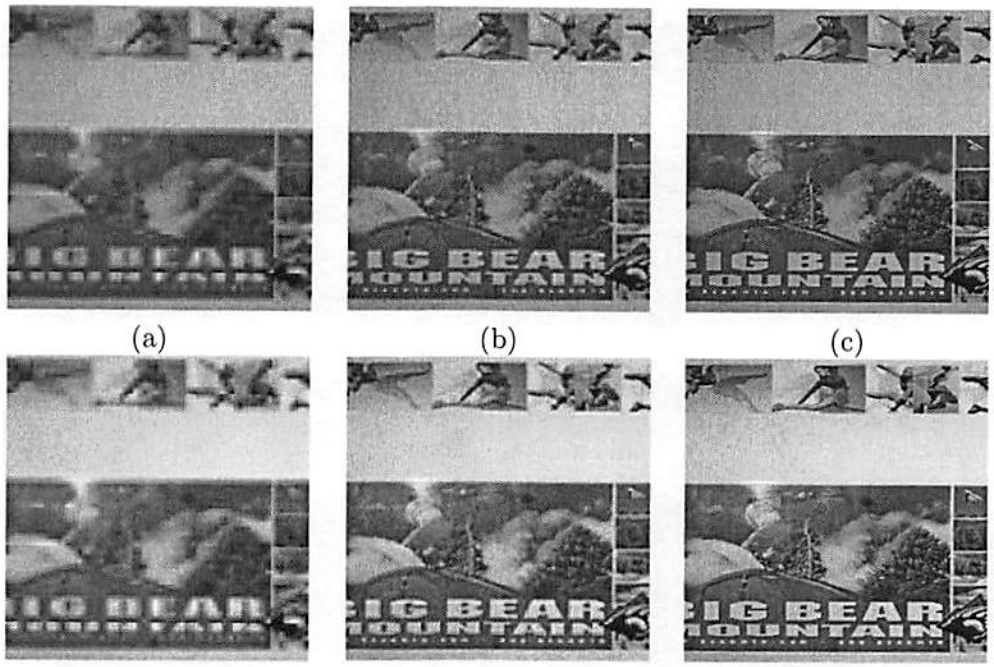


Figure 6.5: Visual comparison of output images with image size of 64×64((a),(d)), 128×128((b),(e)), and 256×256((c),(f)), respectively.

6.1.3 Image Re-sizing

Consider an image that is first downsized by a factor of two horizontally and vertically and then up-sampled back to its original image size. Some content information is lost during this process so that the output image has poorer quality. It is worth to mention that the degradation degree is not all the same throughout the whole image. It actually content-dependent, and it is not efficient to adopt the same processing for the whole image. If the severely degraded areas can be localized, we can focus on enhancing those regions only to improve the visual quality.

The difference between the original HR image and the resized LR image is compared in Fig. 6.6, where the two images are divided into 8×8 blocks and the difference is computed block by block. Then, blocks that have a difference above a certain threshold are marked.

Those blocks are examined for their group type after block classification. Then, we can apply a more advanced processing technique to handle these difficult regions.

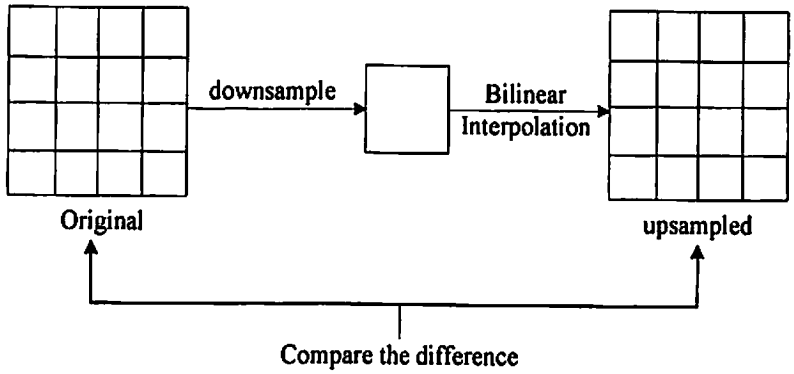


Figure 6.6: Comparison between the original and the degraded images due to image resizing.

We show the original and the resized images in Figs. 6.7 (a) and (b), respectively, where only downsampling/upsampling is considered without any blurring for the resized image. Their absolute difference is shown in Fig. 6.7(c). We see from this figure that the major difference occurs in the edge/texture areas, which is consistent with our expectation. Thus, we only have to focus on the edge/texture blocks for visual enhancement.

According to the above observations, we conclude that the proposed block-adaptive algorithm can save computational complexity while keeping good performance. First, the block-adaptive processing reduces the computational cost efficiently since the degree of freedom is much less than the original image. Second, the visual quality of image-based and block-adaptive methods is close except for regions close to block boundaries. If the boundary can be manipulated carefully, we can improve the image quality at a lower cost. Third, degradation mainly is localized in regions that contain edges and/or textures during the resizing process. Thus, to reduce the complexity more, image enhancement techniques

can be applied to those area rather than the whole image. Due to the above three reasons, the proposed content-adaptive image up-Sampling method provides a good solution that has a good balance between processing complexity and resulting image quality.

6.1.4 Initialization for Block MAP Iteration

As mentioned above, bilinear interpolation is used as the basic image enhancement technique for smooth blocks while the block-based MAP estimator is adopted for blocks that contain edges. To perform the block MAP, we need some initial image for iterative enhancement. Two MAP optimized images initialized by the bilinear interpolation method and the zero-order-hold method are shown in Fig. 6.8 (a) and (b), respectively. We see that the bilinear interpolation method yields a better result in terms of smoothness. The reason is explained by Fig. 6.9. Consider two blocks: an edge block (the green one) and a texture block (the blue one). The MAP method is applied to the edge block while bilinear interpolation is performed on the texture block. If the zero-order-hold method is utilized in the initialization, the expanded matrix will look like the right hand side in the figure. There is some discontinuity between those two expanded blocks in the beginning. Since MAP is performed based on this initialization, the discontinuity tends to exist even after several iterations. Thus, bilinear interpolation should be adopted so that those two blocks has similar behavior in the very beginning.

Although the block-adaptive algorithm provides an efficient way to enhance the visual quality and spatial resolution for edges and textures, the output images contain blocking artifacts due to different processing applied to adjacent blocks. Although the use of bilinear interpolation for initialization helps eliminate the blocking artifact, some of them may still

remain. Thus, a more multi-mode enhancement technique is required for even better quality of output. In other words, instead of dealing complicated and non-complicated cases, the image should be divided into several groups such as plain area, texture, edges and others, so that adaptive enhancement algorithms can be chosen for different needs. For the plain area, an effortless method can be utilized so that the computation cost can be saved for edge blocks to apply more complicated processing. The details and the experimental results are given in the next section.

6.2 Image Up-Sampling with Adaptive Enhancement

In this section, another approach to achieve image quality enhancement is introduced. Again, the DCT-domain block classification is utilized to segment the image into several types, smooth areas, textures, edges and others. Instead of enhancing the quality of edge blocks only, the algorithm introduced in this section is separated into four parts according to the block types.

Blocks that belong to the plain background group contain smooth surfaces. Since there is not much variation in those areas, an effortless zero-order hold method can be adopted to expand the image content without degrading the visual quality much. In contrast, texture can be treated as the spatial repetition of a certain local pattern. Bilinear interpolation followed by a technique called "unsharp masking" [41] are applied to texture blocks to enlarge the block size while magnifying the variations at the same time. This cascaded operation yields an output image block of good quality. The parameters of the unsharp mask, *e.g.* the size of the impulse response array and the weighting coefficients, control the sharpness of the output image. They can be chosen adaptively for different applications.

Since human eyes are more sensitive to edges, upsampling of edge blocks demands special treatment. By viewing an image as a gray-level intensity surface, it can be approximated by a facet model which is built to minimize the difference between an intensity surface and observed image data. The facet model is modified to fit the need of image enhancement which is detailed in the next subsection. For the blocks which are not belonged to these three groups, is categorized as others. The complexity of those blocks falls between the plain background and the edges so that bilinear interpolation is chosen for upsampling whose computation cost is between zero-order-hold method and the facet modeling and unsharp masking.

More details of the key processing, the facet modeling and unsharp masking, which are adopted for image upsampling are given in the following two subsections.

6.2.1 Facet Modeling

As mentioned before, a facet model is built to minimize the difference between an intensity surface and observed image data. The piecewise quadratic polynomial is used in Haralick's facet model [41]. That is, an image $F(j,k)$ is approximated by

$$\begin{aligned}\hat{F}(r,c) &= k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 \\ &\quad + k_7 rc^2 + k_8 r^2 c + k_9 r^2 c^2, \end{aligned} \tag{6.1}$$

where k_n are weighing coefficients to be determined and r and c are the row and column Cartesian indices of image $F(j,k)$ within a specified region. The determination of coefficients k_i, $1 \leq k \leq 9$, demands a least square solution. However, since polynomials $r^m c^n$,

$m, n = 0, 1, 2$, are not orthogonal, the solution of coefficients k_i becomes an ill-conditioned problem. To convert the ill-conditioned problem to a well-conditioned, a set of orthogonal polynomials is used in the polynomial expansion instead. For example, we may consider the use of 3×3 Chebyshev orthogonal polynomials as given below:

$$\begin{aligned} &P_1(r,c) = 1, \quad P_2(r,c) = r, \quad P_3(r,c) = c, \\ &P_4(r,c) = r^2 - \tfrac{2}{3}, \quad P_5(r,c) = rc, \quad P_6(r,c) = c^2 - \tfrac{2}{3} \\ &P_7(r,c) = c(r^2 - \tfrac{2}{3}), \quad P_8(r,c) = r(c^2 - \tfrac{2}{3}) \\ &P_9(r,c) = (r^2 - \tfrac{2}{3})(c^2 - \tfrac{2}{3}), \end{aligned} \tag{6.2}$$

where $r, c \in \{-1, 0, 1\}$. As a result, the approximation can be rewritten in the form of

$$\hat{F}(r,c) = \sum_{n=1}^{N} a_n P_n(r,c), \tag{6.3}$$

where a_n are polynomial coefficients which can be determined by convolving the image with a set of impulse response arrays. To obtain the facet model, we set up observation equations at integer parameters r and c to approximate the image value at a local region. For the image upsampling purpose, we compute $\hat{F}(r,c)$ at non-integer r and c values. It can be used to interpolate an image with any upsampling factor. For example, $\hat{F}(0.5, 0.5)$ can be computed and inserted between $\hat{F}(0,0)$ and $\hat{F}(1,1)$ as shown in Fig.6.10 so that the image size can be enlarged by a factor of two. Similarly, the image size can be adjusted to any desired size by assigning different non-integer parameters such as $(\frac{1}{3}, \frac{1}{3})$, $(\frac{1}{4}, \frac{1}{4})$ into the approximating polynomial.

To compare the performance of bilinear interpolation and facet modeling, some test results are shown in Fig. 6.11. The five input images are a vertical rectangle, a 45-degree triangle, a fan shape, a 135-degree triangle and a horizontal rectangle while the output image are the enlarged version of the input image by a scaling factor of two in each dimension. Fig. 6.11 (a) are images upsampled by bilinear interpolation and (b) are those interpolated using facet modeling. These two methods have similar performance for vertical and horizontal edges. However, for edges with other orientations or curved lines, facet modeling outperforms bilinear interpolation. As compared to the blocky results of bilinear interpolation, facet modeling is capable of capturing the behavior of the edge more accurately so that the output image has smooth edges without annoying artifacts. When other scaling factors are considered, the facet model has additional advantage. That is, the polynomial coefficients are computed only once. To interpolate an image to a different size, we only have to find the proper non-integer r and c values for facet model evaluation. Generally speaking, facet modeling is a good choice to model an edge while dealing with image up-conversion.

6.2.2 Unsharp Masking

An unsharp masking is designed for sharpening an image with edges and details more emphasized. It is commonly used for most digital images due to its applicability for many editing softwares. More details including 2D and 1D unsharp making are given in the following two sections.

6.2.2.1 2-D Unsharp Masking for Texture Blocks

The unsharp masking technique utilizes the information of the blurred version of the original image by convolving with an uniform $L \times L$ impulse response array. After generating the low resolution image, the unsharped masked image $G(j,k)$ can be derived by subtracting the blurred version $F_L(j,k)$ with a certain weighting function from the original image $F(j,k)$, *i.e*,

$$G(j,k) = \frac{c}{2c-1}F(j,k) - \frac{1-c}{2c-1}F_L(j,k), \tag{6.4}$$

where c is the weighting constant and it usually lies in the range $\frac{3}{5} \leq c \leq \frac{5}{6}$. Generally speaking, the sharpening effect gets stronger as c decreases and L increases. An unsharp masking is not capable of adding extra details to the image. Instead, it can enhance the appearance of details by narrowing down the transition band around the edge, *i.e.* increasing the acutance.

As shown in Fig. 6.12, the unsharp mask neither increases the spatial resolution nor transforms the edge into the ideal one (*i.e.* the blue line). However, the image after unsharp masking (*i.e.* the red line) has a larger contrast that results in better visual quality. Note that the 2D unsharp mask considered here has no directional preference, which fits the characteristic of isotropic texture. Therefore, the isotropic 2D unsharp mask is suitable for enhancing the visual quality of isotropic texture. Some examples are given in Fig. 6.13.

6.2.2.2 1-D Unsharp Masking for Edge Blocks

When the edge is taken into account, the situation is slightly different from texture enhancement. Since the edge has an orientation, it is possible to enhance the sharpness of the

edges more efficiently by adopting 1D directional unsharp masking. The mask dimension is reduced to one so that it can be oriented to performed in the direction that is normal to the edge so that the maximum performance is reached. Again, smaller c and larger L provide better performance. However, it requires longer processing time. It is a tradeoff between visual quality and computation complexity.

6.2.3 Experimental Results

In this section, some preliminary experimental results are reported. Test images are of different sizes and with different content complexity. Images are interpolated by bilinear interpolation and the proposed content-adaptive method by a factor of two in each dimension. Since the objective measurement such as MMSE is not able to reflect the visual quality accurately, we show four test results in Figs. 6.14 and 6.15, where images in column (a) are results of bilinear interpolation and those in column (b) are results of the proposed method. It is clear that the proposed algorithm outperforms bilinear interpolation in the resulting visual quality.

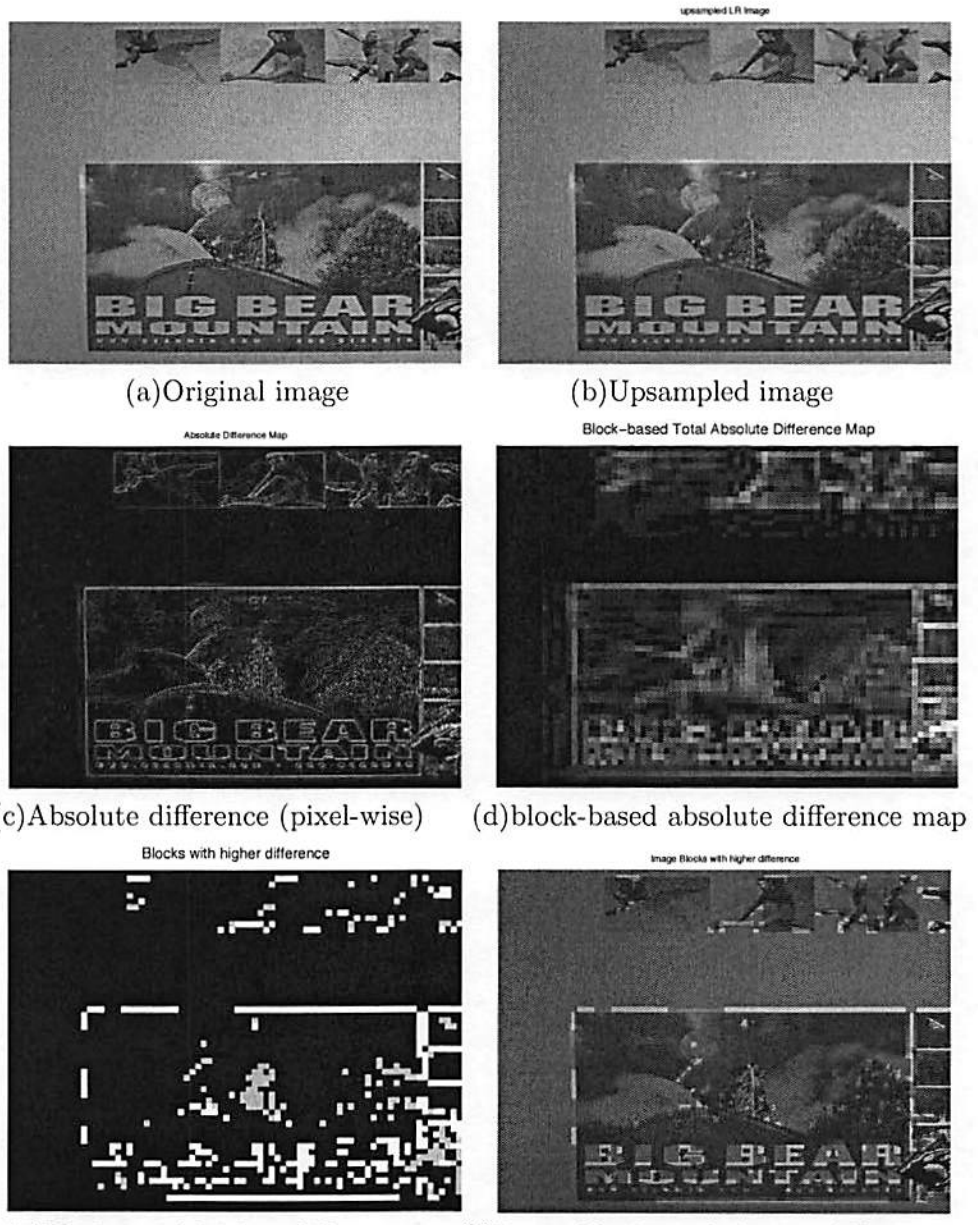
If we zoom in the result by examining the 1D image data across an edge as shown in Fig. 6.16, we see that the line with stars (the proposed method) has a narrower transition band as compare with that of the dashed line (bilinear interpolation). Moreover, the curve of the proposed method has a better match with an ideal curve. Overall, the proposed method has better performance especially in areas that contain edges.

Generally speaking, the proposed method treats an image as a composition of numerous smaller blocks with different contents. From this viewpoint, an image can be categorized into several groups so that adaptive algorithms can be applied more efficiently to different

regions. Experimental results show that the DCT-domain block classification provides fairly good segmentation of image blocks. Although it is not always able to reach 100% accuracy, it behaves as a pre-processing to analyze the image contents to help in algorithmic development to meet various requirements of different applications.

For the process of upsampling, different methods are performed on different areas according to their content complexity. For edges using facet modeling, it has an advantage of flexibility in scaling. It is easy to enlarge an image with any factor by only changing the coordinates without recalculation while the traditional interpolation method may require upsampling followed by downsampling in order to accommodate some desired image size. Furthermore, based on the edge orientation information, the 1D post-processing provides a way to enhance the visual quality even more.

The DCT-domain processing becomes more important for applications nowadays since most image and video are compressed by DCT. In this work, a geometric property inherently in DCT coefficients was investigated and used for block classification. Experimental results show that the proposed block classification using the tree structure works well. The proposed upsampling algorithm based on block classification is content-adaptive which adopts relatively low cost processing for regions that contain less important information to save computational complexity for critical areas that require more sophisticated processing. It was shown by experimental results that the visual quality has been improved with sharper edges and more details in texture areas. How to scale an image sequence efficiently is an interesting topic worth further investigation.



(a)Original image (b)Upsampled image

(c)Absolute difference (pixel-wise) (d)block-based absolute difference map

(g)Blocks with higher difference (h)Image blocks with higher difference.

Figure 6.7: Detecting difference between the original and the resized images using bilinearly interpolation.

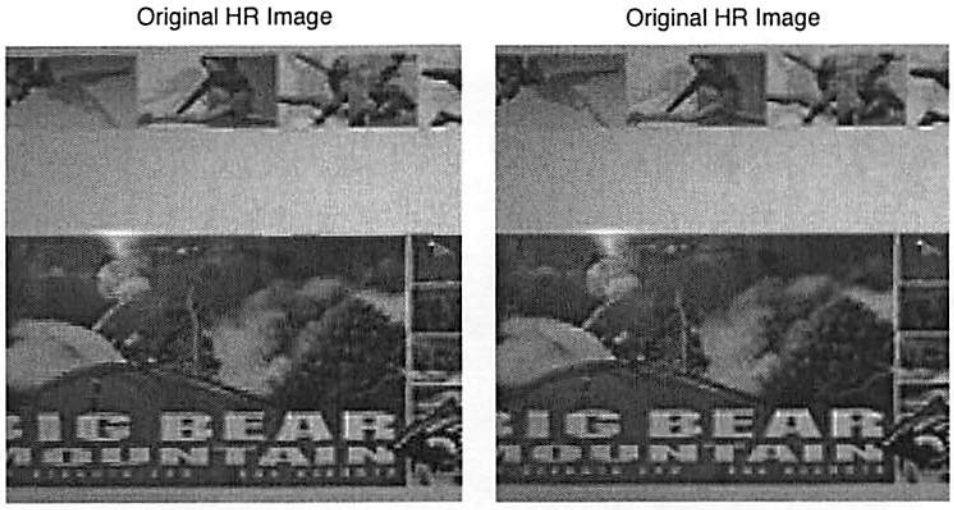


Figure 6.8: The block-based MAP estimator with different initialization methods: (a) zero-order-hold, and (b) bilinear interpolation.

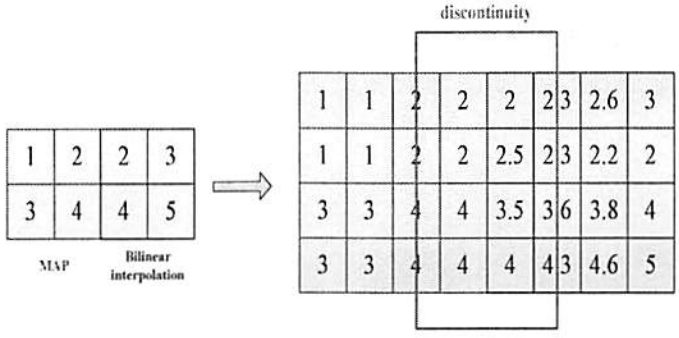


Figure 6.9: Comparison of differences between two initialization methods.

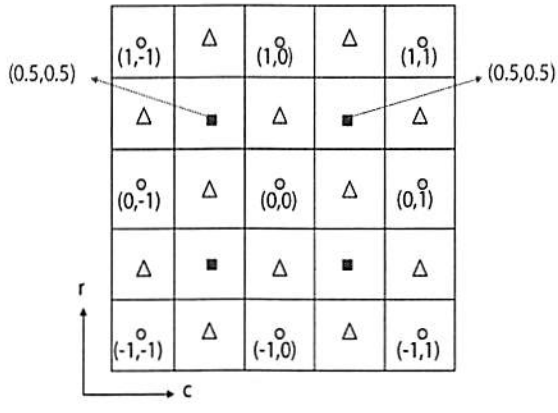


Figure 6.10: The coordinates of a facet model.

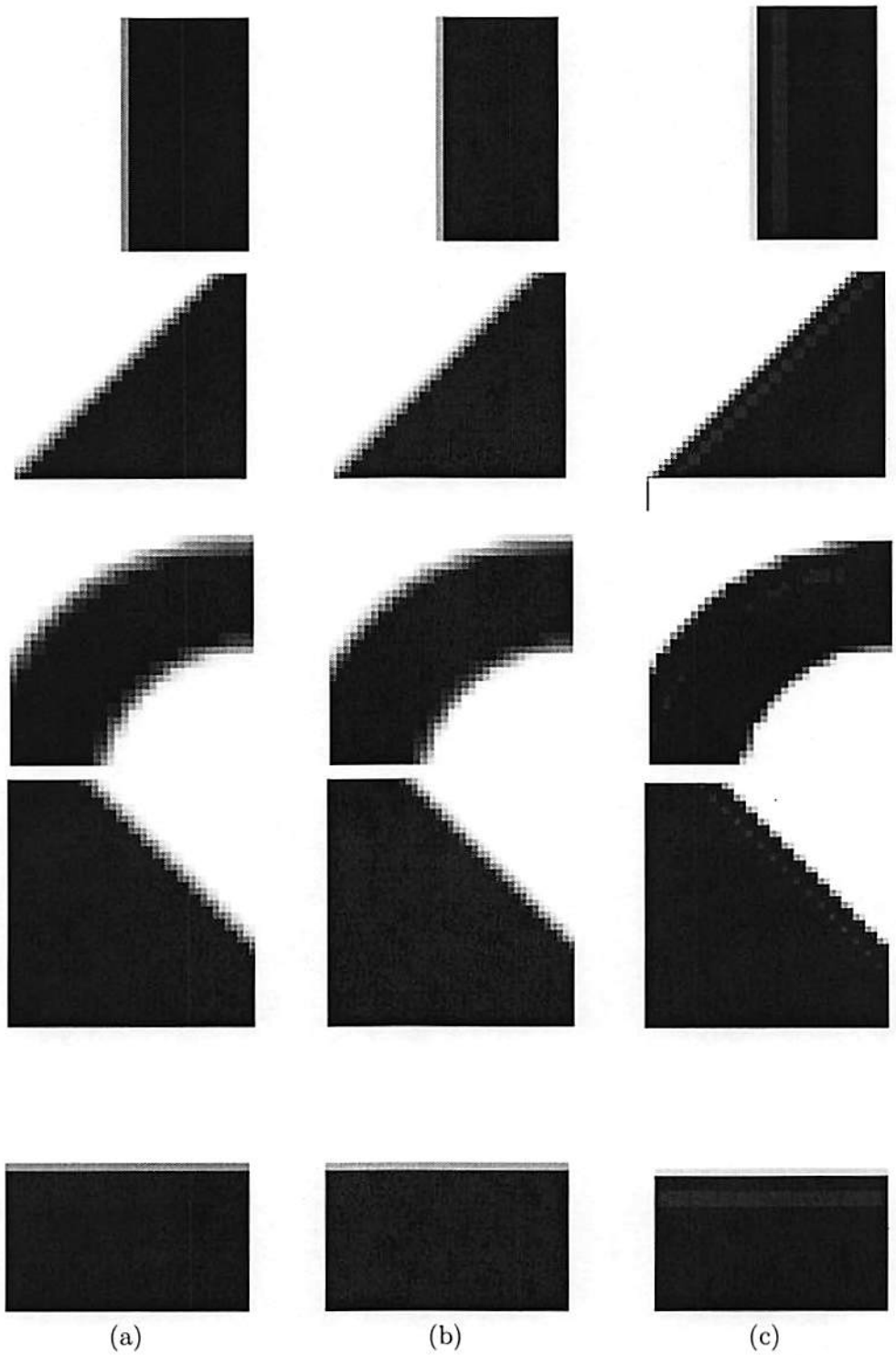


Figure 6.11: Experimental results of (a) bilinear interpolation, (b) the facet model and (c) 1D directional unsharp masking.

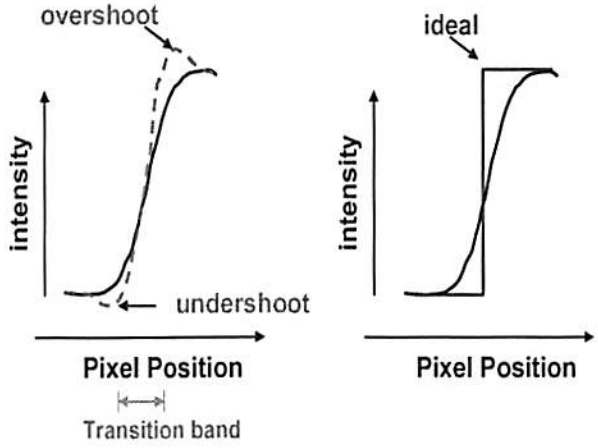


Figure 6.12: Comparison of pixel intensity before and after applying an unsharp mask.

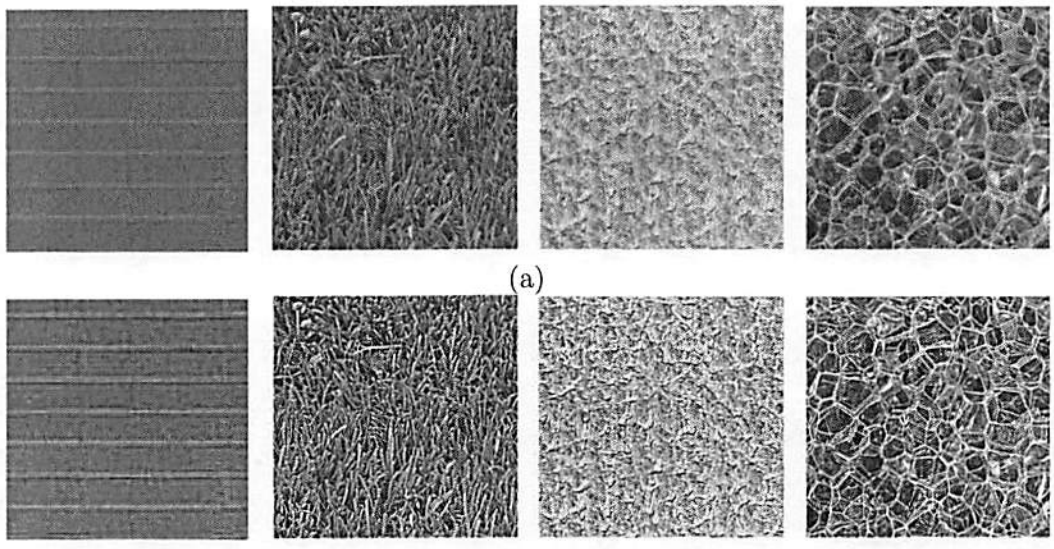
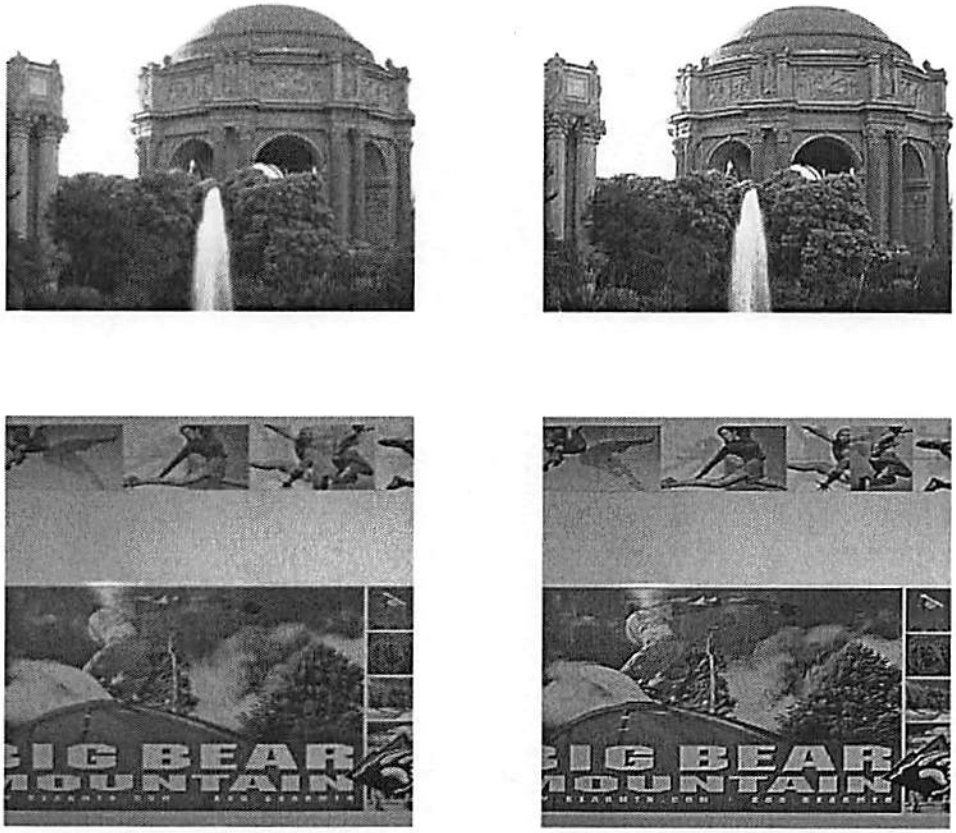
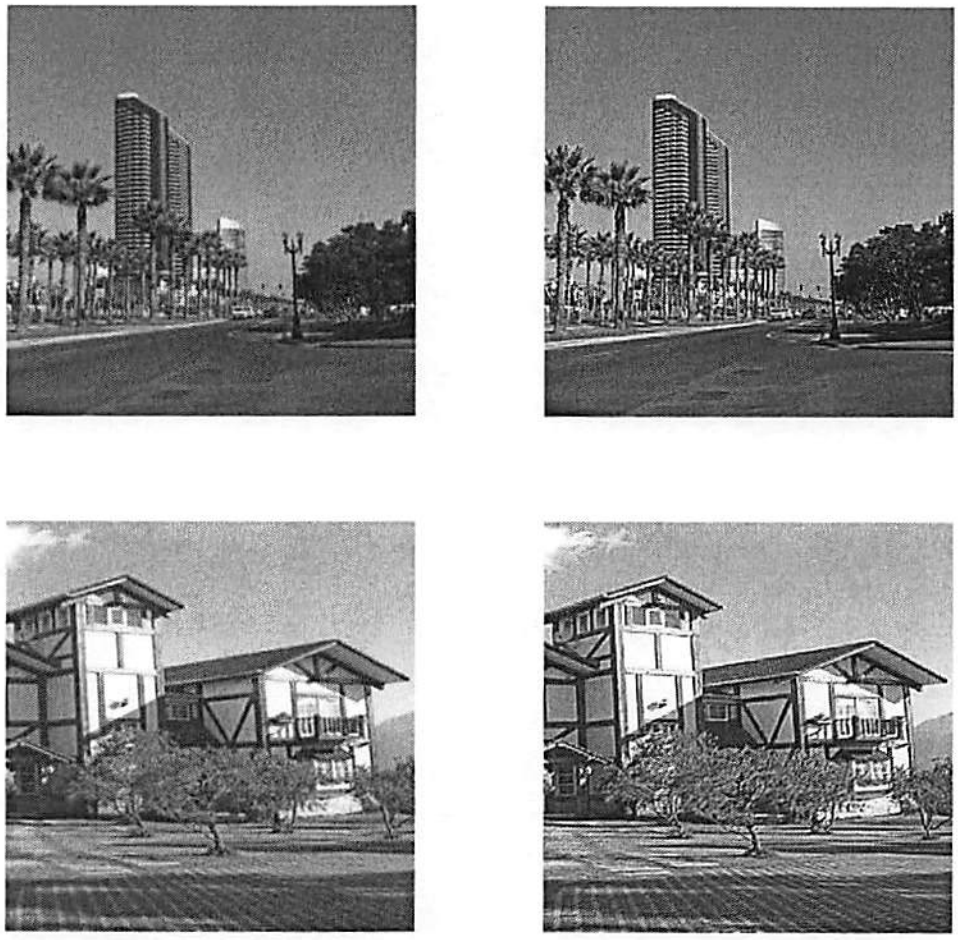


Figure 6.13: Experimental results of unsharp masked texture patterns: (a) the original texture patterns and (b) the unsharp masked texture patterns.



(a) (b)

Figure 6.14: Experimental results of first two test patterns: (a) bilinear interpolation and (b) the proposed content-adaptive upsampling method.



(a) (b)

Figure 6.15: Experimental results of the other two test patterns: (a) bilinear interpolation and (b) the proposed content-adaptive upsampling method.

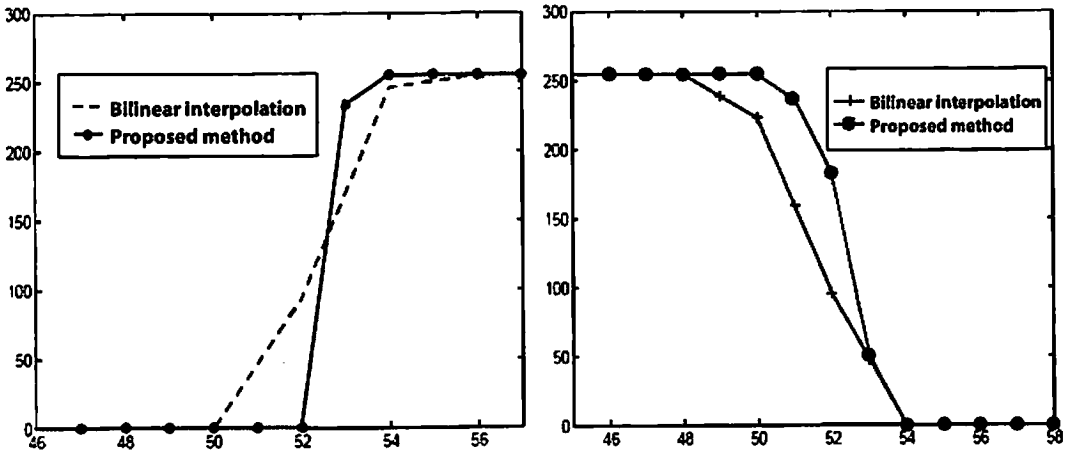


Figure 6.16: The 1D image data across an edge.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The objective of this research is to develop an efficient system to generate an image/video mosaic from multiple image/video inputs captured by different cameras under various conditions. Several techniques were proposed in this work to compensate the color discrepancy and spatial displacement between inputs so as to achieve a high-resolution naturally-looking mosaic output under simplifying assumptions. For example, temporal synchronization and focal length distortion problems are resolved in advance. The developed algorithms are briefly summarized below.

- **Color Matching of Coded Image/Video**

  We considered the problem that two images appear different in their color tones and only have translation displacement between them in Chapter 3. Under the assumption that the overlapping region is well-aligned, we emphasized on matching the color in the compressed domain. We proposed two methods, histogram matching

and polynomial contrast stretching, to compensate the different color tones of two input images in the DCT domain. Both proposed methods are applied only to the DC values. The DC value, which is the average of 64 pixel values within each 8×8 block, represents the average behavior of the block. Therefore, if the original image size is not too small relative to 8×8, the pixel-domain relationship between two images can still be preserved by those DC values in the compressed domain.

It was shown by experimental results that polynomial approximation outperforms histogram matching in terms of output quality and memory requirements. Note that the polynomial used was a second order one. Although higher order polynomials can reduce the bias of matching, it demands more computation and may also result in an increase of variability. This can be demonstrated by our performance evaluation experiments using the MSE measurement, which is defined as the difference between the updated DC values and the expected mean values in the overlapped region. From the experiment, we found that increasing the order does not always lead to performance improvement.

The overlapped regions of two input images were assumed to be well-aligned at the block level, *i.e.* the displacement vector is equal to $[8m, 8n]$ with integers m and n. This assumption is however not practical in real world applications. Thus, an improvement was made to deal with the case where input images have an arbitrary displacement vector of form $[m, n]$. That is, we can perform an interpolation, which computes the pseudo DC value that is located at the well-aligned position. Then, the color matching is applied to those pseudo DC values.

For video color matching, the proposed algorithm can be directly applied to all I frames. In other words, the stretching coefficient is updated for every I frame or every time when a scene change is detected for higher efficiency. For P and B frames, the same procedure can be applied to residuals to obtain another set of parameters. Based on the updated I frame values and the updated residuals of P and B frames, a final estimated value can be computed. For both image and video color matching techniques, only DC values are taken into consideration. Since there is one DC value available for each 8×8 DCT block, the computational cost is reduced down to the scale of $1/64$ as compared with the spatial domain methods. Also, the output of the approximation system is a set of three coefficients which can be stored efficiently and reused for several frames when dealing with image sequences. The proposed color matching technique can produce an image/video mosaic to a satisfactory degree while only a small amount of computation is required. The color matching work was published in [28] and [29].

- **Block-Level Coded Image Registration**

We considered the problem of block-level coded image registration in Chapter 4. For image registration, we assume that the two input images are translated but without any rotation or scaling. Since our target is coded image registration, we consider image registration techniques performed in the DCT domain. We developed two algorithms based on edge estimation and edge detection, respectively.

The method based on edge estimation consists of three steps. First, image segmentation is performed using the DC coefficients of the luminance component for

foreground extraction. Second, for the foreground region, the edge orientation within each 8×8 block is estimated by the DCT coefficients of the first row and first column followed by a 8-level quantization. Finally, a correlation-based technique is performed to find the displacement vector between two images.

The method based on edge detection also consists of three steps: edge detection on the DC map, thresholding and parameter determination. For edge detection, four 3×3 second order edge detectors are applied to the DC coefficients of the luminance component of each input image. Each detector can extract a different edge property so that the generated difference maps preserve edges of various orientations, *e.g.* horizontal, vertical, 45-degree and 135-degree edges. Next, a threshold is set up for each difference map to produce a binary map to filter out some minor edges. Finally, the displacement parameters are determined based on the binary maps of input images generated by the same detector, and the actual displacement vector in the pixel domain is calculated by averaging parameters obtained from all detectors.

It was demonstrated by experimental results that the proposed algorithms saves more than 90% of the computational cost as compared to the traditional pixel domain techniques while the output visual quality remains about the same. The performance is consistent regardless of indoor or outdoor scenes. Although it is a block based processing, the quality of the alignment can be enhanced to the sub-block (4-pixel) accuracy. The results of the coded image registration research were published in [30], [31] and [32].

- **Advanced Coded Image/Video Mosaic Techniques**

  In Chapter 5, we investigated three advanced coded image/video mosaic techniques as summarized below.

  - **Hybrid Block/Pixel Alignment Technique**

    A post-processing technique, called hybrid block/pixel level alignment, was proposed to enhance of the displacement vector resolution from the block level to the pixel level. After applying line detectors to the DC map of an image, the energy of vertical edges of each block can be obtained. Then, a threshold is set to choose the candidates which belong to the group of high energy. For those candidates, a weight is given in order to distinguish them from blocks of other behaviors. The same procedure is performed for labeling blocks of horizontal edges so that a four-value map is available of an image. Several geometric patterns of size 3×3 are predefined for the purpose of determining whether the centered block contains a corner in the spatial domain or not. If a block is classified as a corner block, its eight neighboring blocks and itself are transformed back to the pixel domain for more accurate alignment. As compared with the traditional spatial-domain processing, we do not perform the inverse DCT transform to the whole image but to some selected blocks. It was shown by experiments that the proposed algorithm saves around 40% of the computational complexity while achieving the same quality.

  - **Coded Video Registration**

The problem of stitching two MPEG sequences with a frame rate of 30fps (frames per second) together to become a mosaic video output was investigated. This was done under the assumption that the two image sequences were well aligned in the temporal domain and had the same GOP structure. The proposed algorithm first segments the I frame of each GOP (15 frames in our experiments) into the static background and moving objects. For the static background, the DC values of the luminance component are extracted to form a DC map. Then, based on the DCT domain image registration technique presented in Chapter 4, a set of displacement parameters can be determined. For the moving object, motion vectors are extracted from the remaining frames within the same GOP. Some incorrect motion vectors can be filtered out based on the prior information of the moving object. The displacement parameters can be updated every GOP based on the motion information. It was shown by experimental results that the proposed approach can provide satisfactory performance while keeping the computational low. The video registration results were published in [33].

- **DCT Block Classification**

The DCT domain techniques are attractive since many image and video inputs are of the compressed format using the DCT representation. It is important to analyze the properties of DCT coefficients so that we can bridge the information between the raw and the coded image/video data more conveniently. Since each DCT coefficient represents the energy of a specific pattern with different vertical and horizontal spatial frequencies, we defined some ratio values and developed a tree structure so as to group blocks into different categories based on the

distribution of DCT coefficients in an 8×8 block. It was shown by experimental results that the proposed tree structure can capture some important block types such as the plain background, smooth areas, textures, and edges. Based on the classification result, we can adopt different processing techniques in different areas to save computations.

- **Super Resolution and Image Enhancement**

  The DCT-domain processing becomes more important for applications nowadays since most image and video are compressed by DCT. The geometric property associated with DCT coefficients has been investigated and used for block classification in this work. Experimental results showed that the proposed block classification using the tree structure works well. The proposed upsampling algorithm based on block classification is content-adaptive. That is, it applies the processing techniques of relatively low complexity to regions that contain less important information to save computational complexity for critical areas that require more sophisticated processing. It was shown by experimental results that the visual quality has been improved with sharper edges and more details in texture areas.

7.2 Future Work

The demand on flexible media content conversion across heterogeneous capture and display terminals will continue to grow when more and more terminals are linked by networks. Users will not be only satisfied by rich functionalities of an isolated device but also by

compatibilities between different terminals so that they can get the best output based on the platform available. The difference between terminals has to be compensated by software algorithms to facilitate multimedia data migration from one machine to the other with minimal degradation.

The emphasis will be the balance of computational complexity and resultant image/video quality. Unlike traditional methods, we conduct processing directly in DCT domain and adopting geometric property inherently in DCT coefficients for processing speedup. However, the proposed algorithms have limitations in applicability. More research efforts towards an integrated system that offers flexibility and compatibility among heterogeneous terminals are expected in the near future. Some research issues are highlighted as follows.

- **Eliminating Blocking Artifacts Resulting from Block-based Algorithm**

  In our proposed system, blocks in a whole image frame are classified into several groups by following the tree structure proposed in Chapter 5 based on the distribution of DCT coefficients. Each group has its own specific geometric properties. That is, an image is classified into the plain background, smooth areas, textures, or areas with strong edges or corners. Different geometric properties provide different visual effects. For example, areas with strong edges require better algorithms to improve the resolution since human eyes are more sensitive to those regions. For the areas of the plain background or smooth areas, a simple zero-order-hold method or a bilinear interpolation operation can produce good results. Since blocks are manipulated with different processing techniques individually, there may be artificial block boundaries

generated as a result of block partitioning. Thus, a low complexity post-processing technique is required to remove blocking artifacts.

- **Enhanced Resolution of Moving Objects**

  For multiple video sequences, the regions of interest containing target objects can be combined with their motion information of the following B and P frames for moving object extraction. Based on the movement of the object, we may develop an algorithm especially tailored to enhance the resolution of moving objects. As observed by some researchers, [43], [42], [15], [2], and [16], the quantization step size provides important information about the feasibility of the solution. The estimated solution can be verified using the quantization step size. If the quality of the output video is not satisfactory, some post-processing techniques to further resolution enhancement can be considered. Since we only deal with the regions of interest here, the number of iterations required for the optimal solution is expected to be fewer than that of the traditional iterative approach. Then, the computational cost can be saved while maintaining good performance in visual quality.

Reference List

[1] M.S. Alam, J.G. Bognar, R.C. Hardie, and B.J. Yasuda, "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames," *IEEE Trans. Instrum. Meas.*, vol. 49, pp. 915-923, Oct 2000.

[2] Y. Altunbasak, A. J. Patti, and R. M. Mersereau, "Super-resolution still and video reconstruction from MPEG coded video," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 12, no. 4, pp. 217-226, 2002.

[3] S. S. Beuchemin and J. L. Barron, "The computation of optical flow," *ACM computing Surveys,* vol. 27, pp. 433–467, 1995.

[4] S. Borman, and R. Stevenson, "Spatial Resolution Enhancement of Low-Resolution Image Sequences A Comprehensive Review with Directions for Future Research," *Technical Report, Laboratory for Image and Signal Analysis,* University of Notre Dame, 1998.

[5] S. Borman, and R. L. Stevenson, "Super-Resolution from Image Sequences - A Review," *Circuits and Systems, 1998 Proceedings,* Midwest Symposium, Aug 1998.

[6] N. K. Bose, H. C. Kim, and H.M. Valenzuela, "Recursive Total Least Squares Algorithm for Image Reconstruction from Noisy, Undersampled Multiframes," *Multidimensional Systems and Signal Processing,* vol. 4 no. 3, pp. 253-268, July 1993.

[7] Lisa G. Brown, "A survey of image registration techniques," *ACM Computing Surveys,* (24)4: pp. 325–376, 1992.

[8] Yaron Caspi, and Michal Irani, "A step toward sequence- to-sequence alignment," *CVPR,* 2000.

[9] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," *In Maximum Entropy and Bayesian Methods,* pp. 293-308, Kluwer, Santa Barbara, CA, 1996.

[10] M. Elad and A. Feuer, "Super-Resolution Restoration of Continuous Image Sequence Using the LMS Algorithm," *Proceedings of the 18th IEEE Conference in Israel,* Tel-Aviv, Israel, Mar 1995.

[11] M. Elad and A. Feuer, "Super-Resolution Reconstruction of an Image," *Proceedings of the 19th IEEE Conference in Israel,* pp. 391-394, Jerusalem, Israel, Nov. 1996.

[12] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Processing,* vol. 6 , no. 12, pp. 1646-1658, Dec 1997.

[13] A. T. Erdem, M. I. Sezan, and M. K. Ozkan, "Motion-compensated multiframe wiener restoration of blurred and noisy image sequences, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* vol. 3, pp. 293-296, San Francisco, CA, Mar 1992.

[14] J. Foote and D. Kimber, " FlyCam: practical panoramic video and automatic camera control," *IEEE International Conference on Multimedia and Expo,* vol. 3, no. 30, pp. 1419–1422, Aug, 2000.

[15] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Multiframe resolution-enhancement methods for compressed video," *IEEE Signal Processing Letters,* vol. 9, pp. 170-174, June 2002.

[16] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Super-Resolution Reconstruction of Compressed Video Using Transform-Domain Statistics," *IEEE Transactions on Image Processing,* vol. 13, no. 1, Jan 2004.

[17] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint MAP Registration and High-Resolution Image Estimation Using a Sequence of Undersampled Images," *IEEE Trans. IP,* vol. 6, no. 12, pp. 1621-1633, Dec 1997.

[18] M. Holm, " Toward automatic rectification of satellite images using feature based matching," *Proceedings of International Geoscience and Remote Sensing Symposium,* pp. 2439–2442, 1991.

[19] J. W. Hsieh, H. Y. M. Liao, K. C. Fan, and M. T. Ko, " A fast algorithm for image registration without predetermining correspondence," *Proceedings of the International Conference on Pattern Recognition,* pp. 765–769, 1996.

[20] J.-W. Hsieh, H.-Y. M. Liao, K.-C. Fan, M.-T. Ko, and Y.-P. Hung, "Image registration using a new edge-based approach," *Computer Vision and Image Understanding,* vol. 67, no. 2, pp. 112–130, Aug 1997.

[21] J. Y. Hu and A. Mojsilovic, "Optimal color composition matching of images," *International Conference on Pattern Recognition,* vol. 4, pp. 47–50, 2000.

[22] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, pp. 850–863, 1993.

[23] G. Jacquemod, C. Odet, and R. Goutte, "Image resolution enhancement using subpixel camera displacement," *Signal Processing,* vol. 26, no. 1, pp. 139-146, 1992.

[24] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive Reconstruction of High Resolution Image from Noisy Undersampled Multiframes," *IEEE Trans. ASSP,* vol. 38, no. 6, pp. 1013-1027, 1990.

[25] L. Kitchen, and A. Rosenfeld J.-W., "Gray-level corner detection," *Pattern Recognition Letters.*, vol. 1, pp. 95–102, 1982.

[26] T. Komatsu, T.Igarashi, K. Aizawa, and T. Saito. "Very high resolution imaging scheme with multiple different aperture cameras," *Signal Processing Image Communication,* vol. 5, pp. 511-526, Dec 1993.

[27] L. Landweber, "An iteration formula for Fredholm integral equations of the first kind," *American Journal of Mathematics,* vol. 73, pp. 615-624, 1951.

[28] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "Pixel-and Compressed-Domain Color Matching Techniques for Video Mosaic Application," *Electronic Imaging,* Jan 2004.

[29] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "Color Matching Techniques for Video Mosaic Applications," *ICME,* June 2004

[30] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "DCT-Domain Image Registration Techniques for Compressed Video," *ITCom,* 2004.

[31] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "Compressed-Domain Registration Techniques for MPEG Video," *Electronic Imaging,* Jan 2005.

[32] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "A Fast Compressed-Domain Image Registration Technique for Video Mosaic," *ISCAS,* 2005.

[33] Ming-Sui Lee, Meiyin Shen, C. -C. Jay Kuo, "A DCT-Domain Video Alignment Techniques for MPEG Sequences," *MMSP,* 2005.

[34] H. Li, B. S. Manjunath and S. K. Mitra, "A contour-based approach to multisensor image registration," *IEEE Trans. on Image Processing,* vol. 4, pp. 320–334, 1995.

[35] Aditi Majumder, Gopi Meenakshisundaram, W. Brent Seales and Henry Fuchs, "Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery," *Proceeding of the Seventh ACM International Conference on Multimedia,* October 30 - November 5, 1999.

[36] N. Nguyen and P. Milanfar, "An efficient wavelet-based algorithm for image super-resolution," *Proceedings of International Conference on Image Processing,* vol. 2, pp. 351-354, 2000.

[37] P. Oskoui-Fard and H. Stark, "Tomographic image reconstruction using the theory of convex projections," *IEEE Transactions on Medical Imaging,* vol. 7, no. 1, pp. 45-58, Mar 1988.

[38] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.

[39] A. C. Park, M. K. Park, and M. G. Kang, "Super-resolution Image Reconstruction: A Technical Overview," *IEEE signal processing magazine,* pp. 21-36, May 2003.

[40] A. J. Patti, A. M. Tekalp, and M. I. Sezan, "A New Motion Compensated Reduced Order Model Kalman Filter for Space-Varying Restoration of Progressive and Interlaced Video," *IEEE Transactions on Image Processing,* vol. 7, no. 4, pp. 543-554, Apr 1998.

[41] W.-K. Pratt, *Digital Image Processing,* John Wiley & Sons, Inc., 1978.

[42] M. A. Robertson and R. L. Stevenson, "DCT Quantization Noise in Compressed Images," *Proc. IEEE Int. Conf. Image Processing,* vol. 1, pp. 185-188 2001.

[43] R. R. Schultz and R. L. Stevenson, "Extraction of highresolution frames from video sequences," *IEEE Trans. IP,* vol. 5, no.6, pp. 996-1011, June 1996.

[44] M. Sester, H. Hild, and D. Fritsc, "Definition of ground control features for image registration using GIS data," *Proceedings of the Symposium on Object Recognition and Scene Classification from Multispectral and Multisensor Pixels,* vol. 32, pp. 537–543, 1998.

[45] N.R. Shah and A. Zakhor, "Resolution enhancement of color video sequences," *IEEE Trans. Image Processing,* vol. 8, pp. 879-885, June 1999.

[46] B. Shen and I. K. Sethi, "Direct feature extraction from compressed domain images," *Storage and Retrieval for Image and Video Databases IV,* vol. 2670, 1996.

[47] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "Highresolution Image Reconstruction from Lower-resolution Image Sequences and Space-varying Image Restoration," *ICASSP,* vol. III, pp. 169-172, San Francisco, 1992.

[48] B. C. Tom, A. K. Katsaggelos, and N. P. Galatsanos, "Reconstruction of a high resolution image from registration and restoration of low resolution images," *Proceedings of the IEEE International Conference on Image Processing,* vol. III, pp. 553-557, Austin, TX, 1994.

[49] B.C. Tom and A.K. Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," *Proceedings of 1995 IEEE International Conference on Image Processing,* vol. 2, pp. 539-542, Washington, DC, Oct 1995.

[50] R.Y. Tsai and T.S. Huang, "Multipleframe image restoration and registration," *Advances in Computer Vision and Image Processing,* pp. 317-339, Greenwich, CT:JAI Press Inc., 1984.

[51] M. Tsukada and J. Tajima, "Color matching algorithm based on computational colorconstancy theory," *Proceedings of International Conference on Image Processing,* vol. 3, pp. 60–64, 1999.

[52] A. S. Vasileisky, B. Zhukov, and M. Berger, "Automated image coregistration based on linear feature recognition," *Proceedings of the second Conference Fusion of Earth Data,* pp. 59–66, 1998.

[53] W. H. Wang, and Y. C. Chen, " Image registration by control points pairing using the invariant properties of line segments," *Patern Recognition Letters,* vol. 18, pp. 269–281, 1997.

[54] Z. Zheng, H. Wang, and E. K. Teoh, "Analysis of gray level corner detection," *Pattern Recognition Letters,* vol. 20, pp. 149–162, 1999.

[55] B. Zitova, J. Kautsky, G. Peters, and J. Flusser, "Robust detection of significant points in multiframe image," *Pattern Recognition Letters,* vol. 20, pp. 199–206, 1999.

[56] Barbara Zivota, and Jan Flusser, "Image registration methods: a survey," *Image and Vision Computing 21,* pp. 977–1000, 2003.