

Winter 2007

Blending techniques for underwater photomosaics

Fan Gu

University of New Hampshire, Durham

Follow this and additional works at: <https://scholars.unh.edu/thesis>

Recommended Citation

Gu, Fan, "Blending techniques for underwater photomosaics" (2007). *Master's Theses and Capstones*. 329.
<https://scholars.unh.edu/thesis/329>

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

BLENDING TECHNIQUES FOR UNDERWATER PHOTOMOSAICS

BY

FAN GU

BS, Harbin Institute of Technology, P. R. China, 2005

THESIS

Submitted to the University of New Hampshire

In Partial Fulfillment of

The Requirements for the Degree of

Master of Science

In

Electrical Engineering

December, 2007

UMI Number: 1449586

Copyright 2007 by
Gu, Fan

All rights reserved.

UMI[®]

UMI Microform 1449586

Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ALL RIGHTS RESERVED

©2007

Fan Gu

This thesis has been examined and approved.

Thesis Director, Christian P. de Moustier,
Professor of Electrical and Computer Engineering and
Ocean Engineering

Yuri Rzhanov,
Research Associate Professor of Ocean Engineering

Colin Ware,
Professor of Computer Science and Ocean Engineering

Date

FOREWORD

Two years study in the Center for Coastal Ocean Mapping (CCOM) was a precious period for me. This thesis is almost about all the work that I have done during my stay here.

I owe my greatest thanks to Yuri who has directed and encouraged me for my research. I am also very thankful to Christian and Colin who gave me suggestions and encouragement during my study and helped me with the structure and content of this thesis.

Thanks to my professors: Larry, Lloyd, Jim, Andy, Tianhang, Barbara, Karen, etc. and my labmates: Shachak, Daniel, Lorraine, Val, Mashkoor, Michelle, Briana, Stephen, Luis, Ed, etc., who shared their experiences and knowledge with me. It was really fun to have worked with them.

Of course, I do appreciate the support from my dear parents and my friends in at the UNH: Xin, Yong, Bing, Jing, Mingju, Min, etc..

TABLE OF CONTENTS

FOREWORD	iv
TABLE OF CONTENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
ABSTRACT.....	xiii

CHAPTER	PAGE
1 INTRODUCTION	1
1.1 Challenges	1
1.2 Applications of Underwater Photomosaics.....	2
1.3 Limitations	3
1.4 Previous Work.....	4
1.4.1 Existing Blending Techniques for Underwater Photomosaics	4
1.4.2 Potential Blending Techniques for Underwater Photomosaics	6
1.4.3 Exposure Compensation.....	7
1.5 Contributions from This Thesis	9
2 BACKGROUND TECHNIQUES	11
2.1 Illumination-Reflection Model	11
2.2 Detrending.....	12
2.3 Perspective Mapping.....	14
2.3.1 Perspective Transformation	14

2.4	Levenberg-Marquardt Algorithm.....	16
2.5	Thin-plate Spline.....	17
2.6	Scale Invariant Feature Transform (SIFT).....	18
2.7	Graph-cut Method.....	20
2.7.1	Graph-cut Method vs. Dynamic Programming.....	21
2.8	Poisson Equation.....	22
2.8.1	Image Gradients	22
2.8.2	Solution to the Poisson Equation.....	24
2.8.3	Boundary Conditions	24
2.9	Wavelet Transformations	25
2.9.1	Discrete Wavelet Transformation (DWT)	25
2.10	Objective Evaluation.....	26
2.10.1	Peak Signal-to-Noise Ratio	26
2.10.2	Universal Image Quality Index.....	26
2.10.3	Edge Based Objective Evaluation.....	28
3	PROPOSED BLENDING METHODOLOGY.....	31
3.1	Median Mosaic Based Illumination Correction.....	32
3.1.1	Methodology	33
3.2	Perspective Warping	35
3.2.1	Methodology	36
3.3	Thin-plate Spline Warping.....	37
3.3.1	Methodology	38
3.4	Graph-cut in Gradient Domain	40

3.4.1	A Discussion of Cost Function	41
3.4.2	Methodology	43
3.5	Graph-cut in Wavelet Domain	46
3.5.1	Methodology	46
4	EXPERIMENTAL RESULTS	49
4.1	Median Mosaic Based Illumination Correction.....	50
4.2	Perspective Warping and Thin-plate Spline Warping	55
4.3	Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain.....	57
5	EVALUATION.....	60
5.1	Subjective Evaluation	60
5.1.1	Perspective Warping and Thin-plate Spline Warping	60
5.1.2	Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain	62
5.2	Objective Evaluation.....	63
5.2.1	Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain.....	64
6	DISCUSSION AND CONCLUSIONS	67
	REFERENCES	71
	APPENDICES	77
A	DATA RESULTS	78
A.1	Data Results for Median Mosaics Based Illumination Correction	78
A.2	Data Results for Perspective Warping and Thin-plate Spline Warping	83
A.3	Data Results for Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain.....	89
B	PSEUDOCODE	99

B.1	Pseudocode for Median Mosaics Based Illumination Correction.....	99
B.2	Pseudocode for Perspective Warping	101
B.3	Pseudocode for Thin-plate Spline Warping	103
B.4	Pseudocode for Graph-cut in Gradient Domain.....	105
B.5	Pseudocode for Graph-cut in Wavelet Domain	107
C	PUBLICATION.....	109

LIST OF TABLES

Table 4-1 Comparison of blending methods. The “X” in the table means that the methods are proposed in this thesis, and the rest are shown for comparison.	57
Table 5-1 Components of the test set for perspective warping and thin-plate spline warping.	60
Table 5-2 Components of the test set for graph-cut in gradient domain and graph-cut in wavelet domain.	62

LIST OF FIGURES

Figure 2-1 Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with “X”) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles) [L04]. 19

Figure 2-2 The process of min-cost cut finding. (a) is an overview of the result and (b) is the details of the process. The blocks of $p, q, r, x, y, z, u, v, w$ stands for the overlapping nodes of patch A and B . The edges are assigned values (costs) by Equation (2-33), where the edges that connect the terminal nodes and non-terminal nodes are assigned infinite cost “8”. The min-cost cut is found by connecting the lowest cost edges throughout the overlapping area. 21

Figure 3-1 Framework for Median Mosaicing based illumination correction. The steps are identified in the text. 34

Figure 3-2 Framework for Perspective Warping. The steps are identified in the text. 37

Figure 3-3 Framework for Thin-plate spline warping. The steps are identified in the text. The red and green points represent the features extracted from each frame, and matching points are circled by ellipses represent in the mosaic coordinates. The blue points with in the yellow ellipses are the average of the matching points, and they are identified as the target coordinates for the warping. 40

Figure 3-4 Representation of an overlap situation. The symbols in the figure are given in the text. 42

Figure 3-5 Framework for Graph-cut in Gradient Domain. The steps are identified in the text.....	45
Figure 3-6 Framework for Graph-cut in Wavelet Domain. The steps are identified in the text.....	48
Figure 4-1 Normalized histograms of single image before and after illumination correction. (a) are the original image and its histogram. (b) are the illumination corrected image and its histogram. The horizontal axis of the histogram is the pixel value which ranges from 0 to 255. Each image contains 268×472 pixels.	50
Figure 4-2 Mosaics of 120 frames using methods of averaging (a)-(b), median (c)-(d), closest patch (e)-(f), and graph-cut (g)-(h), without (left column) and with (right column) illumination correction.....	53
Figure 4-3 Back-projected images (frame 5) from the mosaics using different blending methods. Images without (left column) and with (right column) illumination correction, (a) and (b) are original images content and used as the reference images. (c) and (d) are back-projected images from the averaging mosaics. (e) and (f) are back-projected images from the median mosaics. (g) and (h) are back-projected images from closest patch mosaics. (i) and (j) are back-projected images from graph-cut mosaics.	55
Figure 4-4 Comparisons of warping results on one pair of sequential frames (frame 5 and frame 6) with illumination correction. (a) and (b) are from the images without any warping, (c) and (d) are from perspective warping, and (e) and (f) are from thin-plate spline warping. Mosaics in the left column use the feathering method to blend and mosaics in the right column use the graph-cut method to blend.....	56

Figure 4-5 Results of mosaics composed of two sequential frames. (a), (c), (e) are blended using feathering in the spatial domain, the gradient domain, and the wavelet domain; and (b), (d), (f) are blended using the graph-cut method in the spatial domain, the gradient domain, and the wavelet domain.	58
Figure 5-1 Subjective result of perspective warping and thin-plate spline warping.	61
Figure 5-2 Subjective result of feathering, graph-cut in spatial, gradient, and wavelet domains.	63
Figure 5-3 Average PSNR values of the blending methods.	64
Figure 5-4 Average UIGI values of the blending methods.	65
Figure 5-5 Edge based objective evaluation results.	66

ABSTRACT

BLENDING TECHNIQUES FOR UNDERWATER PHOTOMOSAICS

by

Fan Gu

University of New Hampshire, December, 2007

The creation of consistent underwater photomosaics is typically hampered by local misalignments and inhomogeneous illumination of the image frames, which introduce visible seams that complicate post-processing of the mosaics for object recognition and shape extraction. In this thesis, methods are proposed to improve blending techniques for underwater photomosaics and the results are compared with traditional methods. Five specific techniques drawn from various areas of image processing, computer vision, and computer graphics have been tested: illumination correction based on the median mosaic, thin-plate spline warping, perspective warping, graph-cut applied in the gradient domain and in the wavelet domain. A combination of the first two methods yields globally homogeneous underwater photomosaics with preserved continuous features. Further improvements are obtained with the graph-cut technique applied in the spatial domain.

CHAPTER 1

INTRODUCTION

The goals of this thesis are to analyze and compare existing blending methods for underwater photomosaics and to propose several optimal blending methods that are able to compensate for the artifacts existing in the previous methods. The methods are evaluated subjectively and objectively, and the validity of quantitative measures is discussed for underwater photomosaics.

1.1 Challenges

The purpose of underwater photomosaicing is to obtain a visually plausible mosaic with two desirable properties:

- 1) The mosaics should be homogeneous in the appearance of illumination, and without seams.
- 2) The mosaic should retain the features of input images.

While these requirements are widely acceptable for visual examination of a mosaic, their definition as quality criteria was either limited or implicit in previous approaches, thus other statistical measurements are desired.

The objectives of this thesis are to explore the automatic blending methods to alleviate the effects of inhomogeneous illumination, which are always present in the case of artificial lighting, as well as to suppress the visibility of the seams introduced during

the mosaicing to construct an optimal mosaic. Note that the performance of blending techniques may have different meanings depending on the intended user, e.g. scientist trying to deduce large-scale interrelationships; computer program extracting shapes according to some specific rule; or a high-school student learning about the deep-sea environment. It is expected that the mosaics obtained with blending techniques are homogeneous in illumination, continuous, and their features are preserved.

Another challenge of blending is to consider all participating images at once, which is significantly more difficult than dealing with only two overlapping images. Previously, existing blending techniques have been applied to a mosaic of limited size to create composite images and to edit image (for example, removal or insertion of objects in the background). In the case of underwater imagery, mosaic dimensions can rapidly grow in size so that a typical desktop computer cannot handle it. The goal is to get a mosaic of high quality, and the computational complexities and costs of the methods are the secondary consideration.

1.2 Applications of Underwater Photomosaics

In recent years, mosaics created from individual images acquired underwater have been attracting increasing attention from marine geologists, biologists and archaeologists. Optical imaging of the seafloor with submillimeter resolution offers scientists a higher level of detail and ease of interpretation. A variety of oceanographic applications require large area site surveys to study hydrothermal vents and spreading ridges in geology [YBCRW00], benthic ecosystems and species in biology [SERPACT04], and ancient shipwrecks and settlements in archaeology, forensic studies of modern shipwrecks and

airplane accidents [BSMYMWSP02]. These applications can be clearly divided into three categories: First, targeting extraction of quantitative information (for example, geologists need to extract seafloor texture information, the biologists need to extract the shape of the zoolite, and the archeologists need to extract the dimensions of the wreckages); Second, attempting to create a consistent continuous map image, possibly at the expense of minor local distortions, which would provide both the overview and detailed views. The third category that is considered as a special case is recovery of three-dimensional information about the seafloor. This reconstruction method extracts the quantitative information and creates a consistent 3D elevation model. The problem with the third method is that it uses a different approach than the two previous methods and is of substantially higher level of complexity.

1.3 Limitations

A major difficulty to process underwater images is due to the special transmission properties of light in the underwater medium. Light suffers from two processes underwater: 1) absorption, where light intensity reduces, and 2) scattering, a change of direction of the individual photons which is mainly due to the particles of different sizes in the water. In addition, artificial light often used underwater suffers from the difficulties described above, and tends to illuminate the scene in a nonuniform fashion, producing a bright spot in the center of the image with a poorly illuminated area surrounding it.

Due to attenuation and backscatter, light underwater limits the practical coverage of a single image to a few square meters. To obtain larger areas of the scene and compensate for the rapid attenuation of the visible spectrum in water, hundreds or

thousands of images are required and the composite view can only be obtained by exploiting the redundant multiple overlapping images distributed over the scene. However, the non-planar seafloor and the short distance between the seafloor and the camera introduce parallax issues.

Summarizing: Most underwater images are difficult to combine in a consistent mosaic due to limited visibility underwater, use of artificial and spatially inhomogeneous illumination, and parallax issues. In addition, although algorithms for object recognition and shape extraction are typically tolerant of scaling and small distortions, they can be easily confused by feature doubling and rapid changes in illumination. So for the purpose of this thesis, small distortion is allowed, but the seams and ghosting artifacts should be reduced.

1.4 Previous Work

1.4.1 Existing Blending Techniques for Underwater Photomosaics

There are two main types of blending methods for underwater imagery implemented according to their goals: 1) combining the overlapping images to help correct exposure, and 2) only one image is used to preserve the sharpness of features.

The first type attempts to even the exposure to achieve the appearance of homogeneous illumination. The simplest method involves averaging values of all coincident pixels. Feathering (weighted averaging) [PD84] of overlapping pixels is another version of the same approach. It can be used to overcome border effects [SS00]. Another method that executes in the frequency domain is a multi-resolution spline blending [BA83], which has been widely used in different blending applications, since its

publication. The median value method retains the median value of the stack, yielding a smoother mosaic image. In addition, it can be used to get rid of (fast) moving objects [OFT02]. The main deficiencies of the approaches mentioned above are blurring due to mis-registration, or “ghosting” due to parallax effects and moving objects. In case of misalignments between the images, these methods tend to create artifacts in the mosaic image such as double edges.

The other type of blending methods focuses more on the sharpness of features in the mosaics, and it is assumed that by using the patches from only one frame in the mosaic, the features remain more distinctive. The closest patch method [GZBV03] consists of splitting the overlapping images and stitching the geometrically closest patches. The main deficiency of these approaches is that the seams remain visible due to the exposure difference and features that are crossing the boundaries. Another method is to search for a 2D curve in the overlap region along which the differences between two overlapping images are minimal [V99] [BVZ01] [KSE03] [QY05] [GGNM06]. Then each image is copied to the corresponding side of the seam. A variety of approximate optimization techniques have appeared over the years, including simulated annealing [GG84], graph cuts [BVZ01], and loopy belief propagation [TF03]. The graph-cut method combined with the watershed method has already been proposed for this purpose [GGNM06]. The graph-cut technique cycles through a set of simpler alpha-expansion relabeling [BVZ01], each of which can be solved with a max-flow/min-cut polynomial-time algorithm.

1.4.2 Potential Blending Techniques for Underwater Photomosaics

Other potential methods for underwater image blending have also been proposed in image fusion research, such as:

1) The technique named gradient domain stitching [FLW02] [PGB03] [BCVBS01] [LZPW04] [FC88] [ADADCCSC04]. Computation in the gradient domain was recently used for dynamic range compression [FLW02], image editing [PGB03], image inpainting [BSBC00] and separation of images to layers [W01]. In [LZPW04], two approaches were proposed for image stitching in the gradient domain. The closest work is image editing [PGB03], which suggested editing images by manipulating their gradients. One of the editing applications is object insertion, where an object is cut manually from an image, and inserted into a new background image. The insertion is done by solving the Poisson Equation on the gradient field of the inserted object, with boundary conditions defined by the background image. Also, it is similar to photomontage [ADADCCSC04] (the process and result of making a composite photograph by cutting and joining a number of other photographs), which chose different patches from different frame stacks using the graph-cut technique, and then stitching them in the gradient domain.

2) A mosaic blending method by wavelet multi-resolution analysis and variational calculus [SHC01], in which wavelet transformed sub-images at each wavelet space are blended. Variational calculus techniques are applied to balance the image quality between the smoothness around the seam line and the fidelity of the combined image relative to the original images in image blending. A mosaic image is finally obtained by summing the blended images in the wavelet spaces.

3) Warping has also been reported as a method to help correct misalignments due to tilting of the camera and the parallax issues. Perspective mapping as applied in [KJH02] uses point correspondences and rectangle-to-quadrilateral mapping. A thin-plate spline technique [B89] has been used for image warping. Almansa and Cohen proposed the thin-plate spline as a model for the geometric transformations in fingerprint images that is more accurate than the linear transformation [AC00].

1.4.3 Exposure Compensation

Due to the lighting problem underwater, it is necessary to compensate for the brightness difference of every single frame before mosaicing. One existing approach is to estimate a single high dynamic range radiance map from the differently exposed images [MP95][DM97][MN99]. It is assumed in the literature that the input images were taken with a fixed camera whose pixel values are the result of applying a parameterized radiometric transfer function to scaled radiance values. The exposure values are either known or are computed as part of the fitting process. However, this approach has many restrictions for the acquired underwater images whose position information is not accurate and may not work when the camera is simultaneously undergoing large panning motions and exposure changes.

The second approach is to adjust the exposure and contrast of the images by warping their histogram of pixel values [CRH95] [GDC04]. This approach works reasonably well for two frames only, where it is easy to get the average or joint histogram, however, as the number of frames increase and the histograms vary, it becomes difficult to automatically find a template histogram for warping. The average or the joint histogram either lacks contrast or is too noisy to work well. Local histogram

equalization has also been used in the literature [ESH00], and this strategy consists in defining an n by n neighborhood, computing the histogram of this area, and applying an equalization function. However, it is very time consuming and although various algorithms have been devised to make it more efficient, it is still inadequate for real-time applications. Moreover, it has a tendency to amplify noise in areas with poor contrast.

The third approach is detrending, which consists of removing of the trend from images on a heliographic coordinate system. It is based on the assumption that for the imaging model, the image can be decomposed into a reflectance image and an illumination image [HB97]. The low frequency values of the image frame characterize the illumination component which is formed by a slowly varying light field over a smooth surface. The high frequency values of the image frame represent the local contrast. The goal of detrending is to remove the low frequency trend and to preserve the high frequency contrasts.

Techniques that compensate for exposure problems in underwater images [GNC02], include:

- 1) Correcting the acquired image according to a smoothed image estimated through a set of consecutive frames and by disregarding the shade component in the illumination-reflectance model [GW92];
- 2) Local histogram equalization [SHYW98] [ESH00];
- 3) Homomorphic filtering [OSS98] (assuming that the illumination factor varies smoothly through the field of view, this method suppresses the low frequencies while keeping the high frequencies);

4) Subtraction of the illumination field by polynomial adjustment [RLF00] (a low-order 2D polynomial spline is subtracted from the acquired image).

1.5 Contributions from This Thesis

In this thesis, more recent blending methods are explored and modified or combined for usage with underwater images. Results are compared with previous methods objectively and subjectively. Three statistical evaluation methods are explored to test their validity for underwater photomosaics. In order to obtain homogeneity of illumination, a pre-processing detrending is applied, and gradient and wavelet domain techniques are explored. In order to decrease the visibility of seams and obtain a more continuous mosaic, the graph-cut method is used in the gradient domain and the wavelet domain, respectively.

Experiments carried out for this thesis have shown that for underwater images which are often misaligned, graph-cut techniques alone can cause loss of features, especially when features are relatively small. In this case, warping is needed to preserve the features in the mosaics. Geometrical warping of the original images has also been tried here to compensate for the misalignment before using blending techniques. For a start, the perspective mapping method is used as a warping algorithm, where the eight input parameters of this warping method are the coordinates of the four corners of the target image. In order to give more flexibility in the choice of the control points, a thin-plate spline warping method is proposed. The control points are selected by a Scale Invariant Feature Transform algorithm [L04]. Besides obtaining a more distinctive

mosaic, this method will also help to localize features (interest points), which may appear in different locations of the mosaics due to the parallax issue or motion

CHAPTER 2

BACKGROUND TECHNIQUES

In this chapter, the techniques used in this thesis are reviewed in more detail. They include: 1) the general illumination-reflection model, 2) detrending principles, 3) the perspective mapping used to compensate for perspective distortions, 4) an optimization algorithm which is used in perspective mapping, 5) thin-plate spline theories and formulations, 6) a feature detection algorithm which is invariant to image scale and rotation, 7) the graph-cut method and its formulations as an advanced optimal seam selection method, 8) the Poisson Equation, its construction in the digital image gradient and its solution, 9) wavelets transformations, 10) some objective evaluation methods: Peak Signal to Noise Ratio (PSNR), a Universal Image Quality Index (UIQI), and a feature preservation evaluation method based on the edges.

2.1 Illumination-Reflection Model

The digital image is a 2D matrix of pixels with Cartesian coordinates (x, y) , which can be considered as the product of the illumination and reflectance properties of a given scene:

$$F(x, y) = I(x, y) \times R(x, y) \quad (2-1)$$

where $F(x, y)$ is the image luminosity, $I(x, y)$ represents the illumination multiplicative factor, and $R(x, y)$ is the reflectance function or ideal image without shading. The

parameters represent one color channel (red, green or blue). The camera characteristics may also contribute gain $G(x, y)$ and offset $O(x, y)$ terms, so the complete equation can be written as:

$$F(x, y) = G(x, y) \times I(x, y) \times R(x, y) + O(x, y) \quad (2-2)$$

Or, it can be expressed as a reflectance function adjusted by a multiplicative $c_m(x, y)$ and an additive $c_a(x, y)$ shading component:

$$F(x, y) = c_m(x, y) \times R(x, y) + c_a(x, y) \quad (2-3)$$

The illumination factor varies smoothly through the field of view and contributes the low frequency components of the image, whereas reflectance is associated with the high frequency components of the image. In our situation, the $c_a(x, y)$ component is small enough and can be neglected with respect to the multiplicative shading component $c_m(x, y)$. By taking the log of the image, the multiplicative effect is converted into an additive one, allowing the separation of both components:

$$\ln F(x, y) = \ln c_m(x, y) + \ln R(x, y) \quad (2-4)$$

Because the logarithm of the image intensity is the sum of the logarithms of the illumination and reflectance components, a low pass filter or detrending of the log of the image intensity can be applied to reconstruct the illumination component.

2.2 Detrending

Trend in a time series is a gradual change in some property of the series over the whole interval under investigation. It can sometimes be defined as a long term change in the mean, and can also refer to the changes in other statistical properties.

The major methods to estimate the trend are [TFTMM02]:

1. First differencing
2. Digital filtering
3. Piecewise polynomials

In [TFTMM02], the authors compared the methods of linear and quadratic detrending, cubic detrending, wavelet detrending, and spline detrending.

Detrending is the statistical or mathematical operation of removing a trend from the series. It is often applied to remove a feature thought to be distorting or obscuring the relationships of interest.

For a particular frame, let $F_L(x, y) = \ln F(x, y)$ denote the logarithm of the input image, where $F(x, y)$ is the image luminosity. The parametric surface fitting equation for the illumination image can be formulated as:

$$\vec{F}_L \equiv \begin{pmatrix} F_L(0,0) \\ F_L(0,1) \\ \vdots \\ F_L(x,y) \\ \vdots \\ F_L(N,M) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \vdots & & & & & \vdots \\ x^2 & xy & y^2 & x & y & 1 \\ \vdots & & & & & \vdots \\ N^2 & NM & M^2 & N & M & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix} \equiv S\vec{P} \quad (2-5)$$

In the formula above, the estimation order is two, S denotes the surface fitting parameters for each pixel, P is the parameter vector, and (N, M) are the dimension of the image.

The least squares estimate for the parameter vector is

$$\vec{P} = (S^T S)^{-1} S^T \vec{F}_L \quad (2-6)$$

It is noted that the $(S^T S)^{-1} S^T$ term only depends on the order of the trend, and only needs to be computed once.

2.3 Perspective Mapping

Perspective Mapping is realized by using the four corners of two images as the control points (eight degrees of freedom), and obtaining the least square of intensity differences over the overlapping areas using the Levenberg-Marquardt algorithm [SK99]. A simple formation of this warping is to find the perspective transformation between the overlapping region by the normalized correlation and rectangle-to-quadrilateral mapping.

2.3.1 Perspective Transformation

Perspective transformation is a spatial transformation which relates the coordinate system $x = [x_1, x_2]^T$, with $u = [u_1, u_2]^T$ by:

$$x_1 = \frac{a_{11}u_1 + a_{12}u_2 + a_{13}}{a_{31}u_1 + a_{32}u_2 + a_{33}} \quad (2-7)$$

and

$$x_2 = \frac{a_{21}u_1 + a_{22}u_2 + a_{23}}{a_{31}u_1 + a_{32}u_2 + a_{33}} \quad (2-8)$$

This expression can be written in a compact way as:

$$x_1 = \frac{A_1 [u^T, 1]^T}{A_3 [u^T, 1]^T} \quad (2-9)$$

and

$$x_2 = \frac{A_2 [u^T, 1]^T}{A_3 [u^T, 1]^T} \quad (2-10)$$

where

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (2-11)$$

For the 4 corresponding point pairs $[(x_1^i, x_2^i), (u_1^i, u_2^i)], i = 1 \dots 4$, each pair should hold:

$$x_1^i = \frac{A_1[u^{iT}, 1]^T}{A_3[u^{iT}, 1]^T} \quad (2-12)$$

and

$$x_2^i = \frac{A_2[u^{iT}, 1]^T}{A_3[u^{iT}, 1]^T} \quad (2-13)$$

Manipulate the equations to get:

$$A_3[u^{iT}, 1]^T x_1^i - A_1[u^{iT}, 1]^T = 0 \quad (2-14)$$

$$A_3[u^{iT}, 1]^T x_2^i - A_2[u^{iT}, 1]^T = 0 \quad (2-15)$$

Thus, each pair (x^i, u^i) creates a pair of homogenous equation.

Form a vector a from elements of the matrix A :

$$a = [A_1, A_2, A_3] = [a_{11}, a_{21}, a_{31}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}]^T \quad (2-16)$$

Using all known correspondences, the solution can be found from the solution of the over-constrained system of linear equations:

$$A = \begin{pmatrix} u_1^1 & 0 & \dots & u_1^4 & 0 \\ u_2^1 & 0 & \dots & u_2^4 & 0 \\ 1 & 0 & \dots & 1 & 0 \\ 0 & u_1^2 & \dots & 0 & u_1^4 \\ 0 & u_2^2 & \dots & 0 & u_2^4 \\ 0 & 1 & \dots & 0 & 1 \\ -x_1^1 u_1^1 & -x_2^1 u_1^1 & \dots & -x_1^4 u_1^4 & -x_2^4 u_1^4 \\ -x_1^1 u_2^1 & -x_2^1 u_2^1 & \dots & -x_1^4 u_2^4 & -x_2^4 u_2^4 \end{pmatrix}^{-1} \begin{pmatrix} x_1^1 \\ x_2^1 \\ x_1^2 \\ x_2^2 \\ x_1^3 \\ x_2^3 \\ x_1^4 \\ x_2^4 \end{pmatrix} \quad (2-17)$$

2.4 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm (LMA) provides a numerical solution to the mathematical problem of minimizing a function, generally nonlinear, over a space of parameters of the function. It interpolates between the Gauss-Newton algorithm (GNA) [MYF03] and the method of gradient descent [S05].

Its main application is in the least squares curve fitting problem: given a set of empirical data pairs (t_i, y_i) , optimize the parameters p of the model curve $f(t|p)$ to minimize the sum of the squares of the deviations:

$$S(p) = \sum_{i=1}^m [y_i - f(t_i | p)]^2 \quad (2-18)$$

Like other numerical minimization algorithms, the LMA is an iterative procedure. To start a minimization, the user has to provide an initial guess for the parameter vector p . In each iteration step, the parameter vector p is replaced by a new estimate $p + q$. To determine q , the functions $f_i(p + q)$ are approximated by their linearizations:

$$f(p + q) \approx f(p) + Jq \quad (2-19)$$

where J is the Jacobian of f at p .

At a minimum of the sum of squares S , we have $\nabla q S = 0$. Differentiating the square of the right hand side of the equation above and setting it to zero leads to:

$$(J^T J)q = -J^T f \quad (2-20)$$

from which q can be obtained by inverting $J^T J$. The key to the LMA is to replace this equation by a 'damped version':

$$(J^T J + \lambda I)q = -J^T f. \quad (2-21)$$

The (non-negative) damping factor λ is adjusted at each iteration. If reduction of S is rapid, a smaller λ value can be used, bringing the algorithm closer to the GNA. If an

iteration gives insufficient reduction in the residual, λ can be increased giving a step closer to the gradient descent direction. If the iteration number or the reduction of sum of squares for the latest parameter vector p falls short of predefined limits, the iteration is aborted and the last parameter vector p is considered to be the solution.

2.5 Thin-plate Spline

A Thin-plate Spline (TPS) is an interpolation method that finds a “minimum energy” smooth surface that passes through all given points. TPS of 3 control points is a plane, more than 3 is generally a curved surface and less than 3 is undefined.

Given set C of p 3D control points:

$$\left\{ \begin{array}{l} c_{i1} = x_i \\ c_{i2} = y_i \\ c_{i3} = z_i \end{array} \right\}, i \in [1 \dots p] \equiv C_{p \times 3} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_p & y_p & z_p \end{bmatrix} \quad (2-22)$$

solve for unknown TPS weights w and mean of distances between the control points' xy-projection a with a linear equation:

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \vec{w} \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ \vec{o} \end{bmatrix} \equiv L_{(p+3) \times (p+3)} \cdot \vec{x}_{(p+3) \times 1} = \vec{b}_{(p+3) \times 1} \quad (2-23)$$

where K , P and O are submatrices and w , a , v and o are column vectors, given by:

$$K_{ij} = U(|[c_{i1}c_{i2}] - [c_{j1}c_{j2}]|) + I_{ij} \times \alpha^2 \times \lambda, \quad i, j \in [1 \dots p] \wedge \lambda \geq 0 \quad (2-24)$$

$$U(r) = \begin{cases} r^2 \times \log r, & r > 0 \\ 0, & r = 0 \end{cases} \quad (2-25)$$

$$\alpha = \frac{1}{p^2} \sum_{i=1}^p \sum_{j=1}^p |[c_{i1}c_{i2}] - [c_{j1}c_{j2}]| \quad (2-26)$$

$$P_{p \times 3} = \begin{bmatrix} 1 & c_{11} & c_{12} \\ 1 & c_{21} & c_{22} \\ 1 & \vdots & \\ 1 & c_{p1} & c_{p2} \end{bmatrix} \quad (2-27)$$

$$O_{3 \times 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2-28)$$

$$P^T_{ij} = P_{ji}, \quad i \in [1 \dots p] \wedge j \in [1 \dots 3] \quad (2-29)$$

$$\vec{v}_{p \times 1} = \begin{bmatrix} c_{13} \\ c_{23} \\ \vdots \\ c_{p3} \end{bmatrix}, \quad \vec{o}_{p \times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{w}_{p \times 1} = \begin{bmatrix} w_{13} \\ w_{23} \\ \vdots \\ w_{p3} \end{bmatrix}, \quad \vec{\alpha}_{3 \times 1} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2-30)$$

Then interpolate z for arbitrary points (x, y) from:

$$z(x, y) = a_1 + a_2 x + a_3 y + \sum_{i=1}^p w_i U(|[c_{i1}, c_{i2}] - [x, y]|) \quad (2-31)$$

The bending energy (scalar) of a TPS is given by:

$$I_f = \vec{w}^T K \vec{w} \quad (2-32)$$

The locality which refers to the size and shape covering the place of features in Equation 2-25 can be formulated differently for different applications.

2.6 Scale Invariant Feature Transform (SIFT)

The Scale-Invariant feature transformation (SIFT) [L04] is a computer vision algorithm for extracting distinctive features from images. It has been used in algorithms for feature matching and object recognition. The features are invariant to image scale, rotation, and partially invariant to changing viewpoints and illumination.

The major steps in the computation of the image features are [L04]:

1. Scale-space detection:

Search over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function [L04] to identify potential interest points that are invariant to scale and orientation. It is a specific type of blob detection where each pixel in the images is compared to its eight neighbors and the nine pixels each of the other pictures in the scales (Figure 2.1).

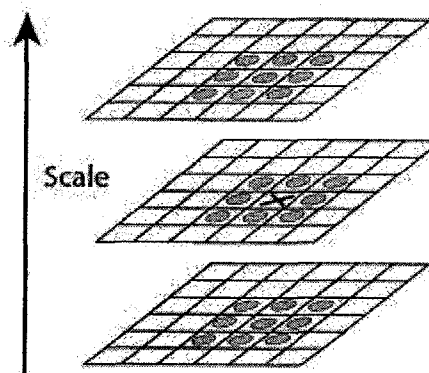


Figure 2-1 Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with “X”) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles) [L04].

2. Keypoint localization:

At each candidate location, a detailed model is fit to determine location and scale. Keypoints are chosen from the extrema in the scale space and selected based on measures of their stability.

3. Orientation assignment:

For each keypoint, in a 16×16 window, histograms or gradient directions are computed. All future operations are performed on image data that has been transformed

relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. Keypoint descriptor:

The local image gradients are measured at the selected scale in the region around each keypoint. The keypoints are represented in a 128-dimensional vector which allows for significant levels of local shape distortion and change in illumination.

2.7 Graph-cut Method

The graph-cut method [BVZ01] has been applied to find an optimal seam between two images so that the seam is the least noticeable. This search is formulated in terms of finding the minimum of a certain energy function. The graph-cut algorithm is based on the principles of combinatorial optimization, and has attracted a lot of attention recently due to its extremely effective ability to solve problems of this type.

Specifically, let x and y be two adjacent pixel positions in the overlap region between two images. Let $A(x)$ and $B(y)$ be the pixel values in the same color channel coming from the original and new images, respectively. The matching quality cost E between the two adjacent pixels x and y that are copied from patches A and B can be defined as:

$$E(x, y, A, B) = \|A(x) - B(x)\| + \|A(y) - B(y)\| \quad (2-33)$$

where $\|*\|$ denotes the selected norm. If an edge between a terminal node (A or B) and a non-terminal node is assigned an infinite cost “ ∞ ”, the non-terminal node will be forced to assume the label from the patch represented by the terminal node. In this case, terminal node refers to the nodes that do not have overlapping correspondences. For example, in

Figure 2-2, both edges e_{Ap} and e_{Bz} have “8” costs, which imply that node p retains its old patch label and node z is assigned to the new patch B . The minimum error path in Figure 2-2 (a) is equivalent to the minimum cost cut of the graph shown in Figure 2-2(b), which can be solved using standard max flow/min cut techniques [BVZ01].

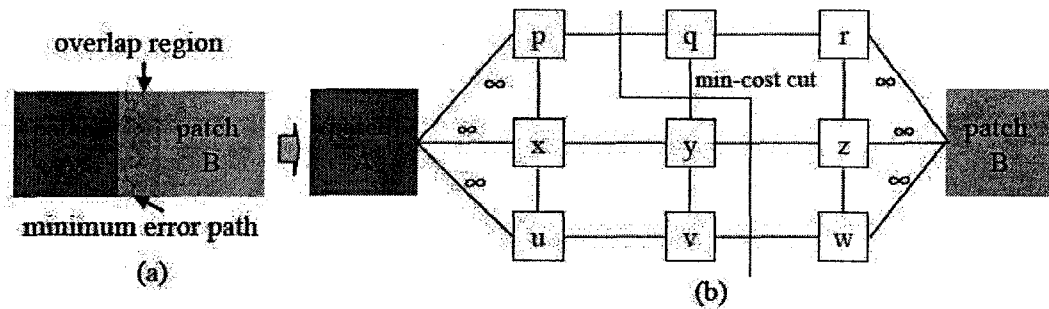


Figure 2-2 The process of min-cost cut finding. (a) is an overview of the result and (b) is the details of the process. The blocks of $p, q, r, x, y, z, u, v, w$ stands for the overlapping nodes of patch A and B . The edges are assigned values (costs) by Equation (2-33), where the edges that connect the terminal nodes and non-terminal nodes are assigned infinite cost “8”. The min-cost cut is found by connecting the lowest cost edges throughout the overlapping area.

2.7.1 Graph-cut Method vs. Dynamic Programming

Similar to the graph-cut technique, the dynamic programming method was first proposed in [EF01], which also makes use of a seam finding process. However, the specifics of implementation impose restrictions on the path that the seam is allowed to follow, which may lead to missing potentially good seams. In addition, when used in the case where images are added one by one, dynamic programming is “memoryless” and cannot explicitly improve existing seams [KSE03]. This causes limitations when appending new images to the existing ones. The graph-cut technique overcomes these

disadvantages by treating each pixel uniformly and is also able to place patches over the existing images in the mosaic.

2.8 Poisson Equation

2.8.1 Image Gradients

Mathematically, the image intensity function is a two-variable function $I(x, y)$ whose gradient at each point (x, y) is given by the derivative of the components with respect to x and y : $\nabla I = (\partial I / \partial x, \partial I / \partial y)$. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. Since the intensity function of a digital image is only known at discrete points, derivatives of these functions cannot be defined unless we assume that there is an underlying continuous intensity function which has been sampled at the pixel locations. With some additional assumptions, the derivative of the continuous intensity function can be computed as a function on the sampled intensity function, i.e. the digital image. For example, gradient for digital images can be implemented as forward differences:

$$\nabla I(x, y) \approx (I(x+1, y) - I(x, y), I(x, y+1) - I(x, y)) \quad (2-34)$$

In order to reconstruct the image, gradient values need to be integrated. In fact, to be integrable, the gradient of a potential function must be a conservative field that satisfies:

$$\frac{\partial^2 I}{\partial x \partial y} = \frac{\partial^2 I}{\partial y \partial x} \quad (2-35)$$

which is rarely the case in our situation.

One possible solution to this problem is to orthogonally project the gradient values onto a finite set of orthonormal basis functions spanning the set of integrable vector fields, such as the Fourier basis functions [FC88]. An alternative approach is to search the space of all 2D potential functions for a function I' whose gradient is the closest in the least-squares sense. In other words, I' should minimize the integral:

$$\iint F(\nabla I', G) dx dy \quad (2-36)$$

where G is the gradient of original image I ,

$$F(\nabla I', G) = \|\nabla I' - G\|^2 = \left(\frac{\partial I'}{\partial x} - G_x\right)^2 + \left(\frac{\partial I'}{\partial y} - G_y\right)^2 \quad (2-37)$$

and G_x and G_y are gradients in the x and y directions.

According to the Variational Principle [W96], a function I' that minimizes the integral in Equation 2-36 must satisfy the Euler-Lagrange equation:

$$\frac{\partial F}{\partial I'} - \frac{d}{dx} \frac{\partial F}{\partial I'_x} - \frac{d}{dy} \frac{\partial F}{\partial I'_y} = 0 \quad (2-38)$$

which is a partial differential equation in I' , and I'_x and I'_y are the gradients in the x and y directions. Substituting F (Equation 2-37) leads to the following equation:

$$2\left(\frac{\partial^2 I'}{\partial x^2} - \frac{\partial G_x}{\partial x}\right) + 2\left(\frac{\partial^2 I'}{\partial y^2} - \frac{\partial G_y}{\partial y}\right) = 0 \quad (2-39)$$

Let $\nabla^2 I' = \frac{\partial^2 I'}{\partial x^2} + \frac{\partial^2 I'}{\partial y^2}$ and $div G = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}$, then the Poisson Equation is

obtained:

$$\nabla^2 I' = div G \quad (2-40)$$

2.8.2 Solution to the Poisson Equation

Since both the Laplacian ∇^2 and div are linear operators, approximating them using the central finite differences yields a linear system of equations:

$$\nabla^2 I = I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4I(x, y) \quad (2-41)$$

and

$$divG = G_x(x, y) - G_x(x-1, y) + G_y(x, y) - G_y(x, y-1) \quad (2-42)$$

There are three methods used to solve the Poisson Equation [PTVWF92]:

1. Direct Method
2. FFT
3. Multi-grid Algorithm

2.8.3 Boundary Conditions

In order to solve a differential equation, one must first specify the boundary conditions. One may consider three cases: Dirichlet boundary conditions, Neumann boundary conditions and mixed boundary conditions.

In [ADADCCSC04], Agarwala et al. have created a Photomontage obtained from different photos taken from the same point of view. They enhanced the overall quality of the results using the Dirichlet and Neumann boundary conditions. When using the Neumann boundary conditions, they set a reference pixel which they call a “pin point” as the level for integration. Then they have used the conjugate gradient. In [LZPW04], Levin et al. solve their problem with an FFT method and Neumann boundary conditions. In [FLW02], Fattal et al. have used a full multi-grid algorithm with Neumann boundary conditions to solve the Poisson Equation. Perez et al. [PGB03] used Dirichlet boundary conditions and a multi-grid method to solve the Poisson Equation.

2.9 Wavelet Transformations

Wavelets are a set of non-linear bases. When projecting (or approximating) a function in terms of wavelets, the wavelet basis functions are chosen according to the function being approximated. Hence, unlike families of linear bases where the same, static set of basis functions are used for every input function, wavelets employ a dynamic set of basis functions that represents the input function in the most efficient way.

2.9.1 Discrete Wavelet Transformation (DWT)

The DWT of a signal x is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse response g resulting in a convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k] \quad (2-43)$$

The signal is also decomposed simultaneously using a high-pass filter h . The outputs give the detailed coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters be related to each other and they are known as a quadrature mirror filter [SA90].

However, since half the sequential frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter outputs are then downsampled by 2:

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k] \quad (2-44)$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n - k] \quad (2-45)$$

2.10 Objective Evaluation

2.10.1 Peak Signal-to-Noise Ratio

Signal-to-Noise Ratio (SNR) measures estimate of the quality of a reconstructed image compared to an original image. In fact, traditional SNR measures do not equate to human subjective perception. The actual metric that has a better performance is peak signal-to-noise ratio (PSNR). It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. First, the mean square error (MSE) of the reconstructed image is calculated follows:

$$MSE = \frac{\sum [I_o(i, j) - I_R(i, j)]^2}{N^2} \quad (2-46)$$

where $I_R(i, j)$ is the reconstructed image pixel and $I_o(i, j)$ is the original image pixel (image contains $N \times N$ pixels). The summation is done over all pixels in the image. The root mean square error (RMSE) is the square root of MSE. Error metrics are computed on the luminance signal only, so the pixel values $I_o(i, j)$ range between black(0) and white (255). PSNR in decibels (dB) is computed by:

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE} \right) \quad (2-47)$$

2.10.2 Universal Image Quality Index

As an outperforming substitute of the standard MSE objective quality measure, the Universal Image Quality Index (UIQI) was proposed in [WB02]. It is designed by modeling any image distortion as a combination of three factors: loss of correlation,

luminance distortion, and contrast distortion. Let $x(i, j)$ and $y(i, j)$ be the original and the test image signals which have N pixels, respectively. The proposed quality index is defined as:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \quad (2-48)$$

Where

$$\bar{x} = \frac{1}{N} \sum x(i, j) \quad (2-49)$$

$$\bar{y} = \frac{1}{N} \sum y(i, j) \quad (2-50)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum [x(i, j) - \bar{x}]^2 \quad (2-51)$$

$$\sigma_y^2 = \frac{1}{N-1} \sum [y(i, j) - \bar{y}]^2 \quad (2-52)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum [x(i, j) - \bar{x}][y(i, j) - \bar{y}] \quad (2-53)$$

The dynamic range of Q is $[-1, 1]$. The best value "1" is achieved if and only if $x(i, j) = y(i, j)$. In a more understandable format, Q can be written as:

$$Q = \frac{\sigma_{xy}}{\sigma_x^2 \sigma_y^2} \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x^2 \sigma_y^2}{\sigma_x^2 + \sigma_y^2} \quad (2-54)$$

The first component is the correlation coefficient between $x(i, j)$ and $y(i, j)$ and its dynamic range is $[-1, 1]$. The second component with a value range of $[0, 1]$ measures how close the mean luminance is between two images. The third component measures how similar the contrasts of the images are, and the dynamic range is $[0, 1]$.

2.10.3 Edge Based Objective Evaluation

Feature is not an exact or universally defined term, and an exact definition often depends on the problem or the application. Usually, a feature is considered to be an “interesting” part of an image, and features are used as a starting point for many computer vision algorithms.

Feature Detection

Typically, features are divided into the following groups:

- Edges: Sets of points in the image which have a strong gradient magnitude.
- Corners (Interest points): Point-like features in an image which have a local two dimensional structure.
- Blobs (Regions of interest or interest points): Blobs provide a complementary description of image structures in terms of regions, but they may also be regarded as interest point operators because they often contain a preferred point.
- Ridges: A ridge detector compute from a gray-level image can be seen as a generalization of the medial axis. However, it is harder to extract ridge features from general classes of gray-level images than edge, corner or blob features.

In image processing and computer vision, the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature or a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions.

Framework of Edge Based Objective Evaluation

The theoretical goal of image blending can be defined [PX05] as: to represent the visual information present in any number of input images, in a single fused image without distortion or loss of information. In practice, the more practical goal of faithful representation of the most important input information in the fused image is usually adopted.

The human visual system (HVS) is more sensitive to sharp edges and details than it is to changes in illumination. Edge information extraction provides a framework for comparison of visual information between the input and blended images.

In this work, the authors used Sobel operator to initially measure the x and y edge components (S^x and S^y) in the input images A , B and the blended image F . Edge parameters are $g_I = (S_I^x{}^2 + S_I^y{}^2)^{1/2}$ and orientation $\alpha_I = \arctan(S_I^y / S_I^x)$, $I \in \{A, B, F\}$. It is assumed that an input edge is perfectly represented if and only if both its strength and its orientation are unchanged in the fused image. When a loss of contrast from A into F exists (an edge in F is weaker than in A), the change in strength, Δ_g^{AF} is defined as the ratio of the fused to the input strength,

$$\Delta_g^{AF}(n, m) = \begin{cases} \frac{g_F(n, m)}{g_A(n, m)}, & g_A(n, m) > g_F(n, m) \\ \frac{g_A(n, m)}{g_F(n, m)}, & g_A(n, m) < g_F(n, m) \end{cases} \quad (2-55)$$

$$\Delta_\alpha^{AF}(n, m) = \frac{|\alpha_A(n, m) - \alpha_F(n, m) - \pi/2|}{\pi/2} \quad (2-56)$$

The quantities Δ_g^{AF} and Δ_α^{AF} describe linear changes in visual information parameters induced by fusion. In order to model the perceived information loss, the

overwhelmingly nonlinear nature of the HVS has to be taken into account. The perceptual loss of edge strength and orientation information with respect to the observed changes in these parameters is modeled as:

$$Q_i^{AF}(n, m) = \frac{\Gamma_i}{1 + \exp\{-\kappa_i[\Delta_i^{AF}(n, m) - \sigma_i]\}} \quad (2-57)$$

where $i \in \{g, \alpha\}$, and parameters $\kappa_g, \kappa_\alpha, \sigma_g, \sigma_\alpha$ are set according to their effects on the edge information preservation [PX05], Γ_g and Γ_α are set to insure $Q_g^{AF} = 1$ and $Q_\alpha^{AF} = 1$ when Δ_g^{AF} and Δ_α^{AF} are equal to 1.

The combined preservation measurement can be expressed as:

$$Q^{AF}(n, m) = [Q_g^{AF}(n, m)Q_\alpha^{AF}(n, m)]^{1/2} \quad (2-58)$$

Edge preservation images that represent local success of information fusion are formed between each of the inputs and the fused output, Q^{AF} and Q^{BF} . The overall success of fusion of images A and B into F, $Q^{AB/F}$ is then obtained as a normalized sum of local edge preservation Q^{AF} and Q^{BF} weighted by their respective perceptual importance w_A and w_B :

$$Q^{AB/F} = \frac{\sum_{n=1}^N \sum_{m=1}^M Q^{AF}(n, m)w_A(i, j) + Q^{BF}w_B(i, j)}{\sum_{i=1}^N \sum_{j=1}^M w_A(i, j) + w_B(i, j)} \quad (2-59)$$

CHAPTER 3

PROPOSED BLENDING METHODOLOGY

In this chapter, five blending methodologies for underwater photomosaicing are proposed. The goal is to produce a mosaic satisfying the requirements: 1) preservation of images with maximum sharpening for recognition, 2) seamless mosaics with homogeneous illumination. Most traditional methods can only reach one of these goals, for example, the median mosaicing, feathering, or multi-resolution spline can give a seamless mosaic with homogeneous illumination, however, most of the features are blurred or have “ghosting” artifacts. The optimal seam methods can preserve the sharpness of the features, however, due to the illumination difference, the seams are unavoidable. Methods, such as the optimal seam in the gradient domain try to take care of both problems, but are not appropriate in underwater applications because of the irregular distribution of underwater photomosaics.

In the previous chapters, artificial lighting was mentioned as one of the major problems that cause inhomogeneous illumination. To solve this problem, a trend correction of the input frames is proposed in the first section of this chapter. I found that due to the misalignment of the images, no matter what blending method is used, artifacts always exist due to the differences in the overlapping region. For the optimal seam methods, these misalignments affect the seams. In other words, the results depend on the order in which the frames are added. In the second section, geometric warping techniques

are used: perspective warping and thin-plate spline warping to correct the errors caused by misalignment. In the third part, the methods of graph-cut in gradient and wavelet domain are proposed to achieve simultaneous homogeneous illumination and feature preservation in mosaics.

It should be noted that, the methods proposed in this thesis assume that the registration for mosaicing have been achieved, with the Fourier-based featureless procedure, or the feature-based one. In the latter case, the tolerance value used in robust matching is much smaller than the threshold in the former method. The misalignment mentioned in this thesis refers to the artifacts caused mainly by parallax.

3.1 Median Mosaic Based Illumination Correction

In order to get a continuous mosaicing with unnoticeable seams, many blending methods utilize fusion with various weighting functions, in the spatial or the frequency domains [BA83]. An alternative solution is to change the exposure of images before blending, using the radiometric correction, histogram warping, and detrending. Different assumptions and applications may require different approaches. For underwater images, which do not hold the assumption that the camera parameters are easy to estimate, and which are complex in the histogram estimation, the radiometric correction and histogram warping methods are difficult to apply.

Practically, the image can be decomposed into illumination and reflection components, so the illumination component can be corrected separately. In this section, the detrending technique is implemented for underwater photomosaicing, and median

mosaicing is used as the trending reference. This method takes advantage of the quadratic detrending method which has moderate complexity but is efficient.

Median mosaicing has been widely used for underwater photomosaicing. It takes the median value from the image stack that is composed of overlapping pixels. Unlike simple averaging, it is not affected by outliers. Although the median mosaic is blurred due to its low pass filter characteristics, the overall illumination of median mosaicing is more homogenous. Hence the illumination trend of the median mosaic can be used to correct the illumination inhomogeneity of the original frames.

Here, calculations are based on the logarithm of the input image which converts the multiplicative operators to additive. By fitting a parametric surface to estimate the illumination image, the reflectance image can be constructed. This method is similar to the homomorphic filtering [OSS98], but the advantage here is that it is less sensitive to local intensity variations such as shadows and backscatter.

3.1.1 Methodology

In [RLF00], the authors proposed to use detrending to process single frames before blending. The method proposed here is to use the trend obtained from the median mosaic, and a second-order fitting polynomial. The median mosaics yield global consistency for the illumination. The method proposed here is illustrated in Figure 3-1 with the following steps:

- 1) Perform median mosaicing on original video frames.
- 2) Back-project the corresponding images of the frames from the median mosaic.
- 3) Obtain the trend of back-projected frames and original frames based on the surface fitting parameters in log space (Equation 2-5).

- 4) Warp the trend of the original frame to match the trend of the corresponding back-projected frame.
- 5) Use the trend-corrected image for mosaicing.

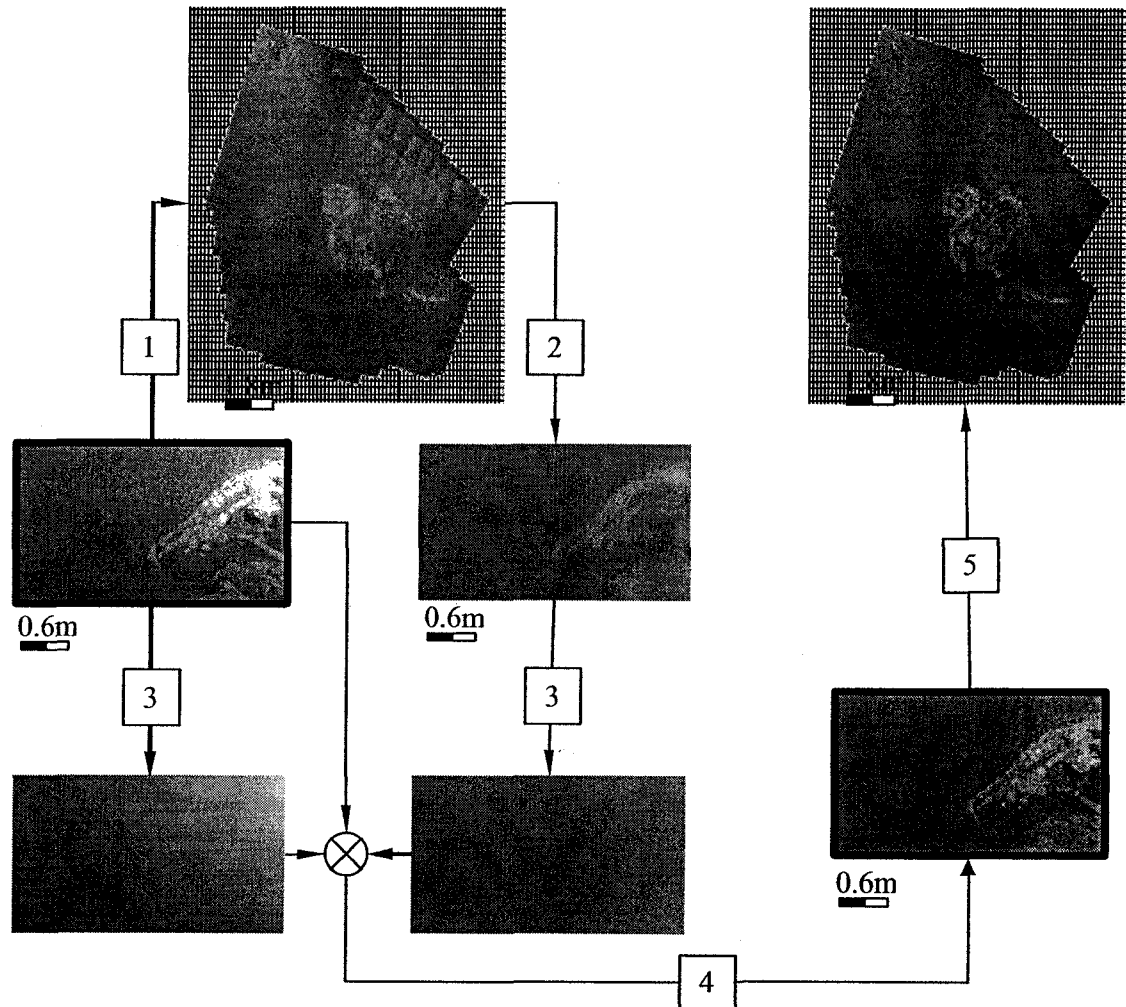


Figure 3-1 Framework for Median Mosaicing based illumination correction. The steps are identified in the text.

3.2 Perspective Warping

Underwater images have to be acquired at a short range a few meters, so that the assumption of planarity is almost never satisfied. This explains the choice of a rigid affine model in this thesis, which is substantially simpler than the perspective model.

The seams caused by the illumination inhomogeneities can be diminished by illumination correction, but due to misalignment of frames, blurring artifacts remain in results produced by previous methods. In addition, local misalignments make the result of the graph-cut method used in this thesis dependent on the order in which individual frames are added to the mosaic. The comparison of the original frame and back-projected image shows that relative small features are easily “cut off” in the mosaicing by the graph-cut techniques. Consequently, further improvements are needed to help solve this problem.

Image warping is a branch of image processing that deals with geometric transformation techniques and it has benefited greatly from several fields, ranging from early work in remote sensing to recent developments in computer graphics. Warping in mosaicing consists of choosing of a set of control points and corresponding shift vectors determining final locations of these control points.

Perspective warping consists of choosing image corners as the control points and attempting to shift them around [KJH02]. Thus this becomes an optimizational problem by finding shift vectors that minimize some objective function describing differences between overlapping images (in this thesis, this function is defined as a normalized sum of squared differences between pixel values contributing to the same location in a mosaic coordinate). This approach is identical to choosing the perspective model for describing

camera motion. The negative side of perspective warping is that its brightness constancy constraint, which rarely holds for underwater images.

3.2.1 Methodology

The perspective warping steps taken are similar to the work done in [KJH02], and are illustrated in Figure 3-2 for two overlapping images I_B and I_D :

- 1) From the registration records, find the world homography of the two images in the mosaic.
- 2) Fix the coordinates of four corners of I_B , and perspective warp I_D (Equation 2-7 and Equation 2-8) until the sum of the square of the differences between the two images in the overlapping area comes to the minimum, obtaining I'_D .
- 3) Mosaic the warped image I'_D and the fixed images I_B .

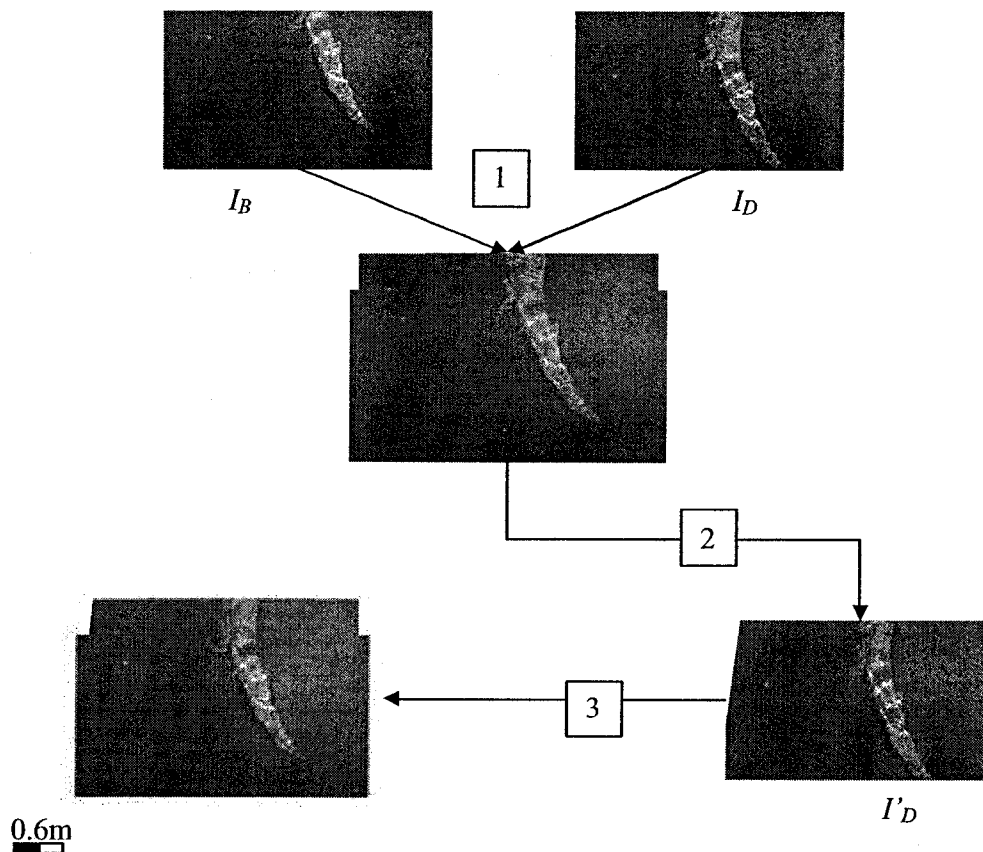


Figure 3-2 Framework for Perspective Warping. The steps are identified in the text.

3.3 Thin-plate Spline Warping

Blurring of features has the most impact on mosaic quality the most and suggests choosing the most prominent features of the frames as control points. The number of these features is typically much larger than four per image [B89]. On the other hand, feature-based warping allows for formulation of the problem in terms different from optimization and does not rely on the brightness constraint for images.

Thin-plate Spline (TPS) is an interpolation method that finds a “minimally bended” smooth surface that passes through all given (“control”) points. When used for image warping, it moves the control points to the target positions. In the perspective

warping discussed above, the control points are four corners, and it only deals with the perspective distortion.

The thin-plate spline warping method proposes to select more control points locally, and move them to the position where they will coincide in the mosaic. At the same time, the neighborhood of the control points is interpolated according to the locality. In the control point selection, this thesis employs the Scale Invariant Feature Transform (SIFT) algorithm. This algorithm extracts distinctive features from images and has been widely used in feature matching and recognition. The features are considered to be invariant to image scale, rotation and partially invariant to changing viewpoints and illumination.

3.3.1 Methodology

The proposed algorithm is illustrated in Figure 2-3 with the following steps:

- 1) Points of interest are extracted from two overlapping frames using the SIFT algorithm.
- 2) Two sets of points are matched using local invariant point descriptions with the additional constraint that matching points must not be separated in the mosaic by a distance larger than pre-determined threshold. Note that the matching stage does not require use of any robust procedure like RANSAC (RANDOM SAMPLE Consensus) [FB81].
- 3) For all matched pairs, a new location of control points (features) is placed at a mid-distance between the original locations in the mosaic-based coordinates, these new locations are found in the space of the original images, thus determining shift vectors for all control points.

- 4) Thin-plate spline warping is applied to participating images. The size of images is left unchanged, so some data may disappear, and some pixels may be marked as having no data.
- 5) Warped images are mosaiced again, with the same transformation as before. Features chosen as control points are mapped onto the same mosaic location.

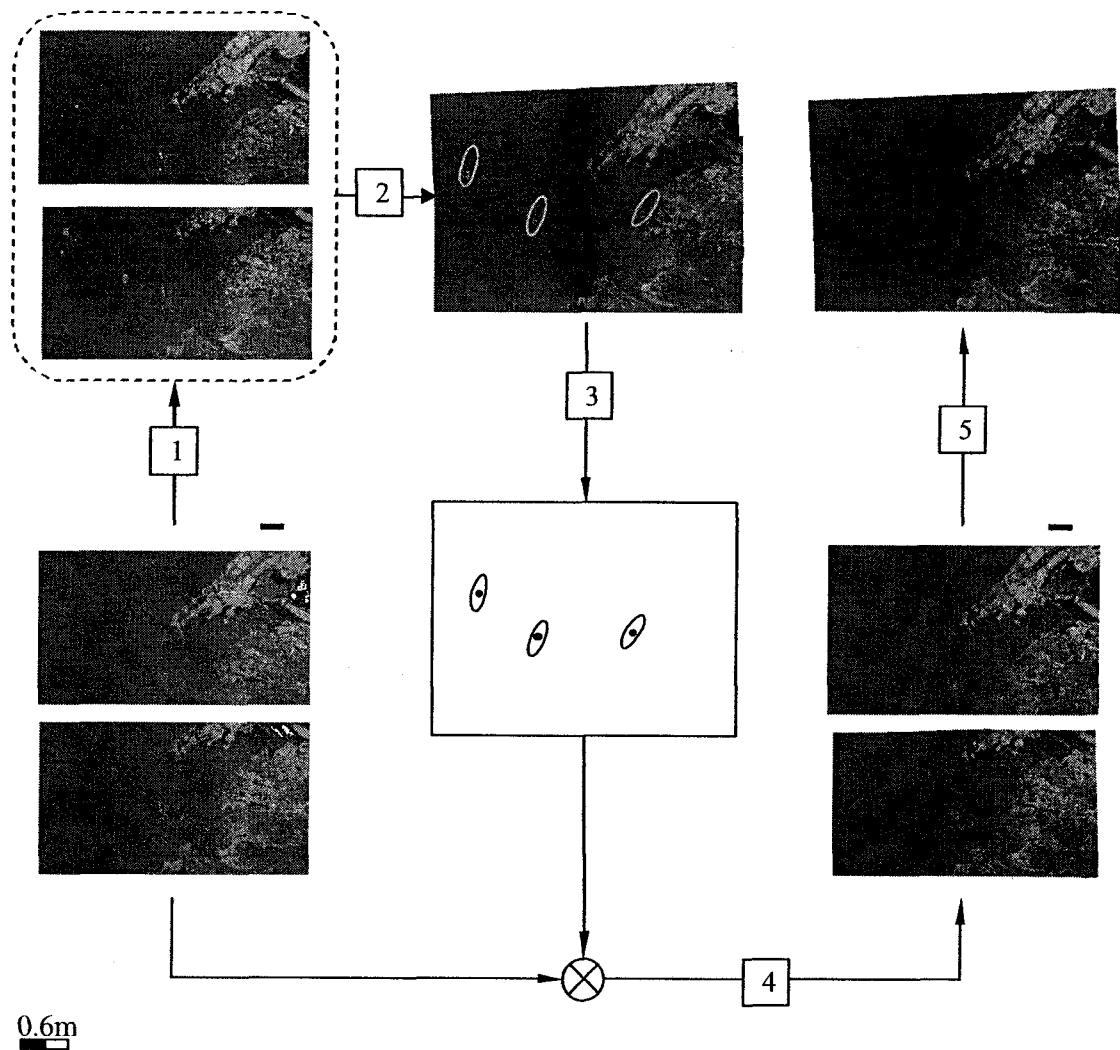


Figure 3-3 Framework for Thin-plate spline warping. The steps are identified in the text. The red and green points represent the features extracted from each frame, and matching points are circled by ellipses represent in the mosaic coordinates. The blue points within the yellow ellipses are the average of the matching points, and they are identified as the target coordinates for the warping.

3.4 Graph-cut in Gradient Domain

Blending methods that combine all the pixels from a stack of images in a region of overlap usually suffer from “blurring” effects. An efficient method to suppress

blurring and preserve the sharpness of features is to find some optimal curve or seam in the overlap region and copy only the pixels of each patch into the corresponding side. However, it is difficult in practice to achieve a pleasing balance between homogeneous illumination and preservation of sharp transitions to prevent blurring.

Gradient domain techniques [FLW02] [PGB03] [BCVBS01] [LZPW04] [FC88] [ADADCCSC04] have been reported as an efficient method to avoid both artifacts. In image editing [PGB03], the insertion boundary is expected from a single background image, whereas in this case, two images are added together and the boundary from both images should be considered. In addition, unlike what is done in Photomontage [ADADCCSC04], the boundary in our case is not rectangular and there is no single “pin pixel” (the fixed pixel that determined the color level of the final mosaic) in solving the Poisson Equation (Equation 2-40).

3.4.1 A Discussion of Cost Function

In the graph-cut in gradient domain method, the blending quality in the seam region is measured in the gradient domain. The mosaic image should contain a minimal amount of seam artifacts, i.e. a seam should not introduce a new edge that does not appear in individual images. For image dissimilarity, the gradients of the mosaic image are compared with the gradients of overlap images. This reduces the effects caused by global inconsistencies between the blended images. Specifically, the blended image is computed by minimizing a cost function E_p , where E_p is a dissimilarity measure between the derivatives of the blended image and the derivatives of the input images. Let I_1 and I_2 be two aligned input images; τ_1 and τ_2 be the region viewed exclusively in image I_1 and I_2 , respectively; and ω be the overlap region, as shown in Figure 3-4,

with $\tau_1 \cap \tau_2 = \tau_1 \cap \omega = \tau_2 \cap \omega = 0$. Let W be a mask image. The blending result \hat{I} is defined as the minimum of E_p with respect to \hat{I} :

$$E_p(\hat{I}; I_1, I_2, W) = d_p(\nabla \hat{I}, \nabla I_1, \tau_1, W) + d_p(\nabla \hat{I}, \nabla I_2, \tau_2, U - W) \\ + d_p(\nabla \hat{I}, \nabla I_1, \omega, W) + d_p(\nabla \hat{I}, \nabla I_2, \omega, U - W) \quad (3-1)$$

Here, U is a uniform image, and

$$d_p(J_1, J_2, \phi, W) = \sum_{q \in \phi} W(q) \|J_1(q) - J_2(q)\|_p \quad (3-2)$$

with $\|\cdot\|_p$ as the l_p -norm.

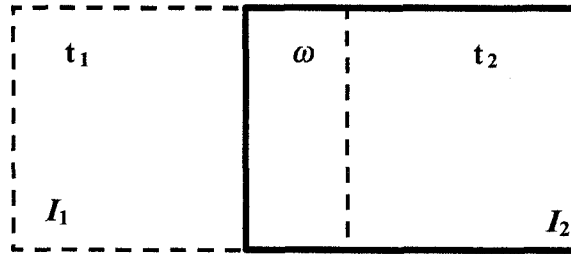


Figure 3-4 Representation of an overlap situation. The symbols in the figure are given in the text.

The first two terms in the right-hand side of Equation 3-1 express the dissimilarity of the mosaic image to the input images in the respective regions. Dissimilarity in the gradient domain is invariant to the mean intensity of the image. In addition, it is less sensitive to smooth global differences between the input images, e.g. due to non-uniformness in the camera photometric response and due to scene shading variations. The last two terms in Equation 3-1 contain the gradients in the overlap region, penalizing the derivatives that are inconsistent with any of the input images. In image locations where

both I_1 and I_2 have low gradients, these terms penalize high gradient values in the mosaic image. This property is useful in eliminating false blending edges.

The choice of norm has implications on both the optimization algorithm and the mosaic image. It has been proven [LZPW04] that the l_1 -norm and the optimal seam method give the same result when there is a consistent seam between the input images; results under l_2 -norm are equivalent to feathering of the gradient images followed by a solution of the Poisson Equation (Equation 2-40).

3.4.2 Methodology

The method proposed in this section is designed to overcome the defects of: 1) the single graph-cut technique, which result in apparent seams when two images have inconsistent inhomogeneous illumination, and 2) the simple gradient domain stitching, which can still cause blurring in a misaligned case. As mentioned in the previous sections, the proposed method executes the graph-cut in the gradient domain. Regarding boundary conditions, it is straightforward to modify the frame being added according to the existing mosaics [FY06].

The procedure is illustrated in Figure 3-5. Assuming that the two images have already been aligned and taking only one color channel for illustration, the steps are:

- 1) From the registration records, find the world homography of the two images in the mosaic.
- 2) According to the overlapping area (which in general is an irregular polygon), identify two rectangle patches that have the minimum coverage of overlapping areas.
- 3) Calculate the difference of overlapping areas.

- 4) Perform graph-cut on the difference.
- 5) Calculate the gradient field of two rectangle patches.
- 6) Combine the two gradient fields according to the mask.
- 7) The spatial values are reconstructed by solving the Poisson Equation with Dirichlet boundary conditions.
- 8) The corresponding reconstructed values are put back in the final mosaic.

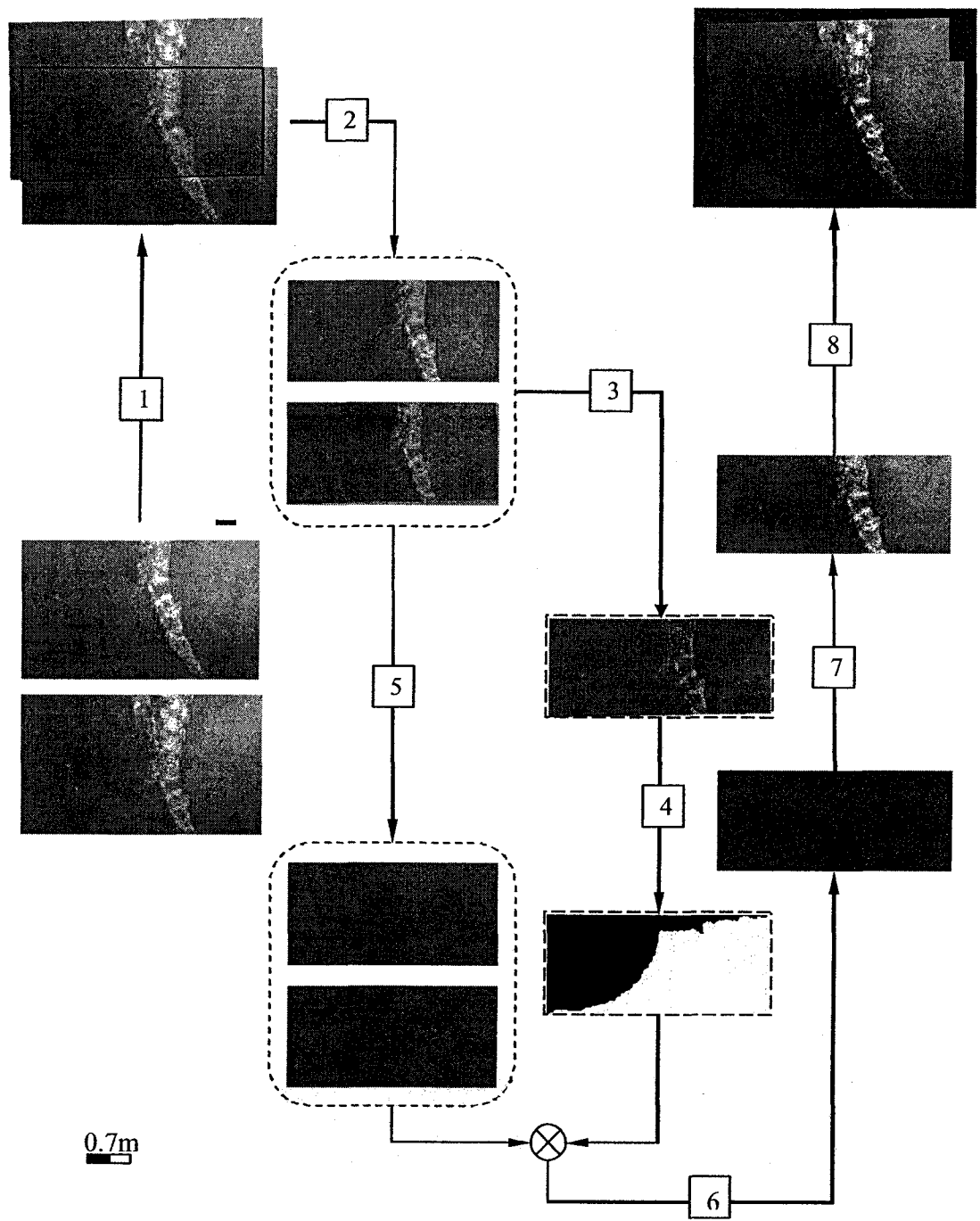


Figure 3-5 Framework for Graph-cut in Gradient Domain. The steps are identified in the text.

3.5 Graph-cut in Wavelet Domain

Since wavelets were first introduced in the graphics community, they have come a long way and are now an important tool in many graphics and image processing applications. Performing operations in the wavelet domain has the following advantages [DL01]:

- Multi-resolution
- Progressive computation
- Space-frequency locality
- Compatibility with emerging standards

The blending technique realized in the wavelet domain has also shown its effectiveness. Iddo and Dani [DL01] gave a demonstration of this application. However, their algorithm of combining coefficients of sub-bands to compensate for illumination differences is only applicable when multiple images are well matched. Blurring effects could occur if images do not match. In underwater photomosaicing, even the most effective registration algorithm cannot guarantee that exact matching could be achieved.

3.5.1 Methodology

The wavelet transform (WT) is a mathematical tool that can be used to describe 1D or 2D signals in multiple resolutions. A wavelet transform is obtained through a sequence of low-pass and high-pass filters, alternated with down-samplings. The result of the wavelet transform is a down-sampled smoothed signal and several detail coefficients obtained at each down-sampling. In other words, the wavelet transform produces a signal that encodes both information on the original signal values and its multi-scale edges. The

algorithms used here are based on the space-frequency locality and multi-resolution properties of the wavelet.

The idea of these algorithms come from the basic principle of multi-resolution spline blending [BA83]; that is, performing multiple operations on different frequency sub-bands to smooth the transition zones. It is consistent to the rules of Human Visual System that human eyes are more sensitive to the sudden changes in the image. The difference here is that my models are wavelet based, while [BA83] is using the Laplacian pyramid. When the process comes to the transition zone, the technique described in [BA83] used the method of feathering on the overlap region in different sub-bands, which is not practical in our case because of the occurrence of blurring.

The procedure is illustrated in Figure 3-6. Assuming that the two images have already been aligned and taking only one color channel for illustration, the steps are:

- 1) From the registration records, find the world homography of the two images in the mosaic.
- 2) Referring to the overlapping area (which in general is an irregular polygon), identify two rectangle patches which have the minimum coverage of overlapping areas.
- 3) Perform wavelet transformation on two rectangle patches.
- 4) Obtain the difference wavelet values of the two patches.
- 5) Perform the graph-cut technique in the wavelet sub-bands respectively and obtain the graph-cut masks.
- 6) Multiply the masks and the corresponding wavelet sub-bands, and obtain the final wavelet coefficients.

7) Transform back into the spatial domain and put it back into the mosaic.

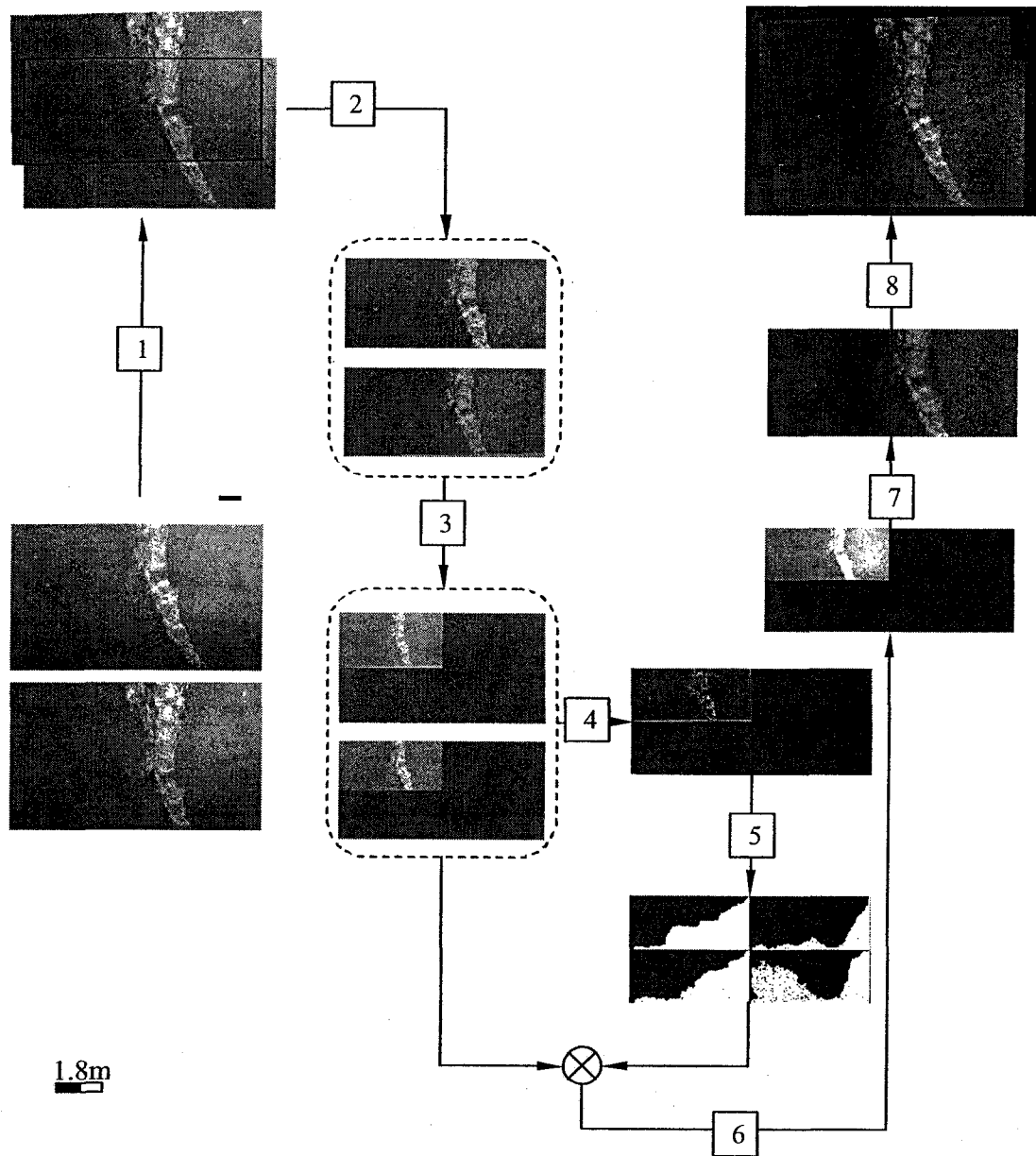


Figure 3-6 Framework for Graph-cut in Wavelet Domain. The steps are identified in the text.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, experiments are described that evaluate the methods proposed in the Chapter 3. The data used in this thesis is from Monterey Bay Aquarium Research Institute (MBARI), collected in Monterey Bay using Remotely Operated Vehicle (ROV) Tiburon. The imaged region covers an area about $7 \times 11 \text{m}^2$ and contains 120 frames, each frame is approximately $2 \times 3 \text{m}^2$ extracted from a video sequence.

The purpose of median mosaic based illumination correction is to obtain a mosaic that appears to have homogeneous illumination with reduced seams. The histogram of the original frame and the corrected frames are compared. The effectiveness of this illumination correction is evaluated by comparing mosaics of 120 frames obtained using four methods: (1) averaging, (2) median, (3) closest patch, and (4) graph-cut.

The purpose of perspective warping and thin-plate warping is to preserve the features. The performance of these warping techniques is evaluated on two sequential frames which are corrected using median mosaic based illumination correction. The mosaics are obtained using: (1) feathering (weighted averaging), and (2) graph-cut.

The purpose of the graph-cut in the gradient domain and graph-cut in the wavelet domain methods is to obtain homogeneous mosaics with preserved features. Results are presented from two sequential frames. In order to compare the property of feature preservation, feathering (weighted averaging) in gradient domain [LZPW04] and wavelet domain is performed.

4.1 Median Mosaic Based Illumination Correction

The median mosaic based illumination correction method proposed in Chapter 3 intends to reduce the illumination inhomogeneity due to the artificial lighting underwater. The normalized histograms of the original image and illumination corrected image are shown in Figure 4-1. The pseudocode used to implement this illumination correction is given in APPENDIX B.1.

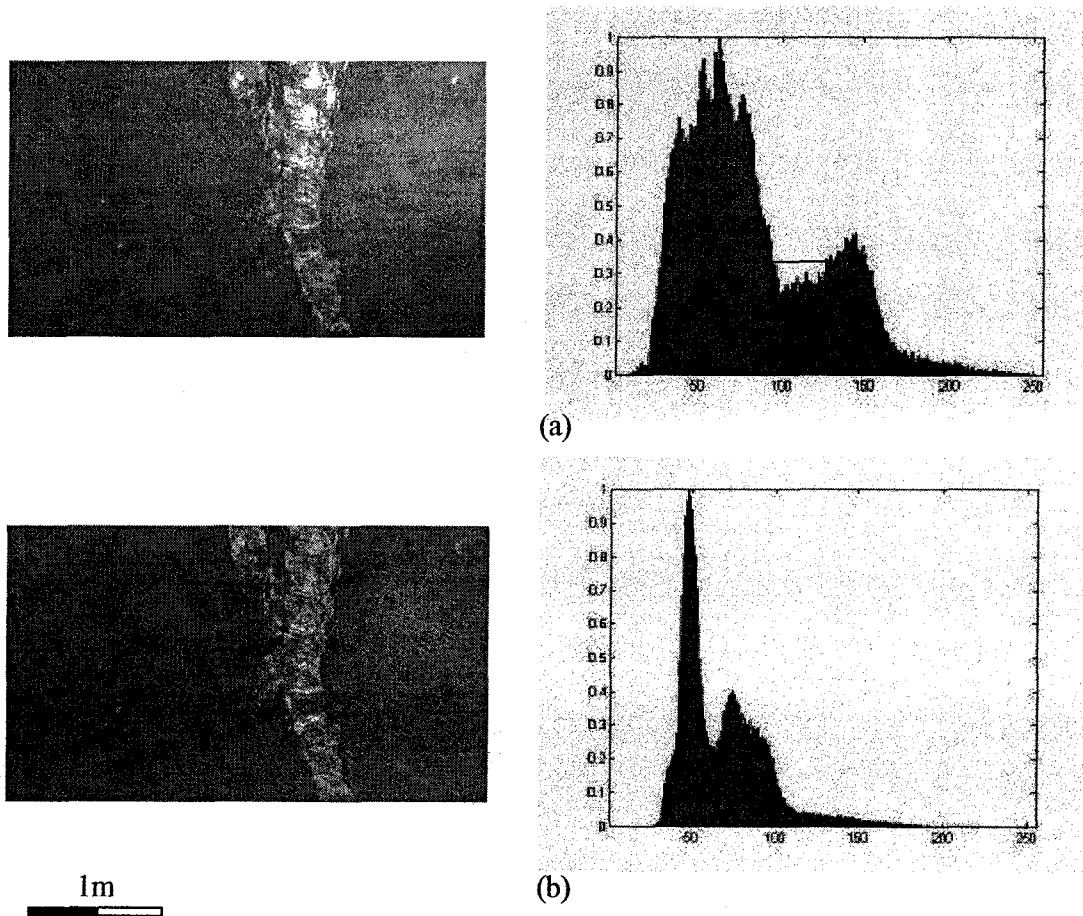
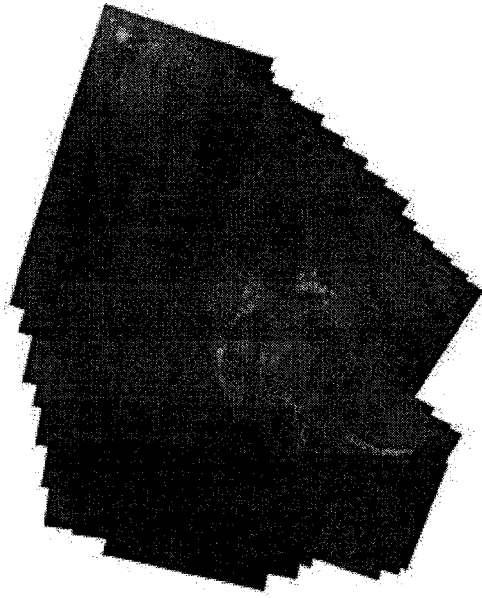


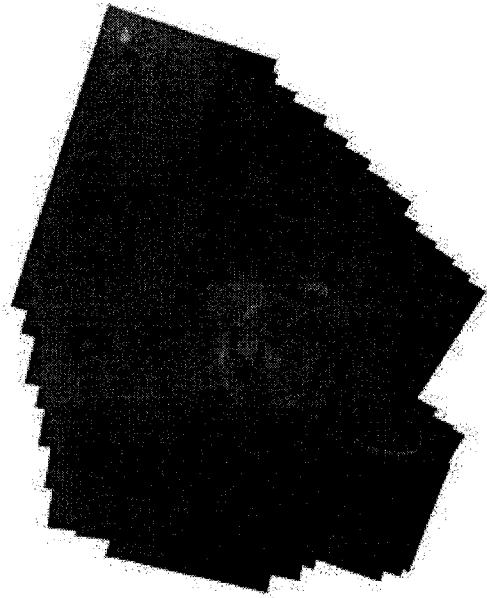
Figure 4-1 Normalized histograms of single image before and after illumination correction. (a) are the original image and its histogram. (b) are the illumination corrected image and its histogram. The horizontal axis of the histogram is the pixel value which ranges from 0 to 255. Each image contains 268×472 pixels.

The normalized histograms show that the range of pixel values is shorter after illumination correction. It means that the values are more concentrated and thus more homogeneous in illumination.

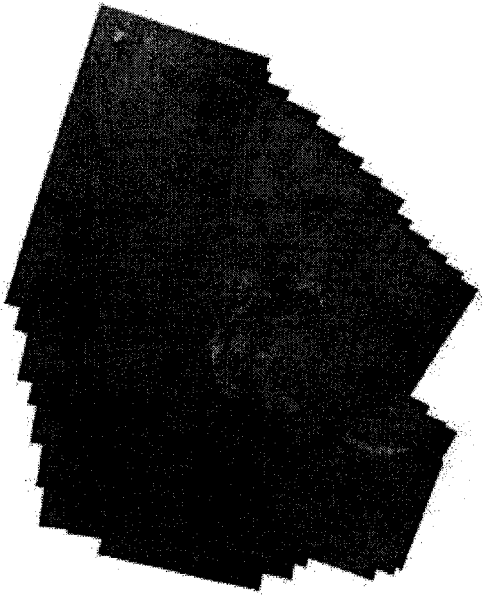
The comparison in this section is based on four existing blending methods: (1) averaging, (2) median, (3) closest patch, and (4) graph-cut. In Figure 4-2, the mosaics of 120 frames are given. In Figure 4-3, the single back-projected images (according to the footprint of frame 5 in the mosaic) from Figure 4-2 are extracted and compared. In APPENDIX A.1, an additional four back-projected images are given. Back-projected image are referring to the image obtained from final mosaic according to its world transformation record.



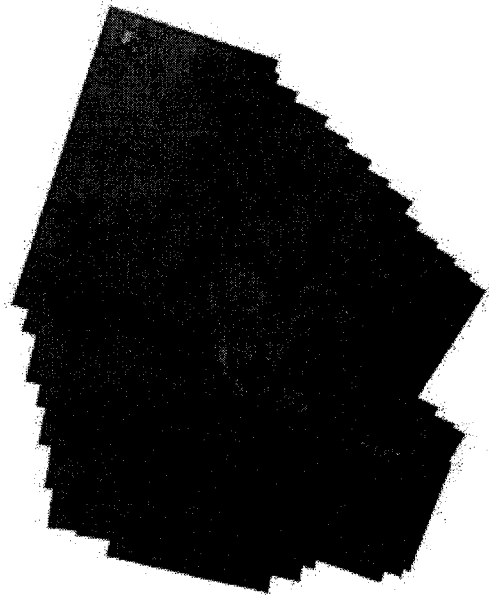
(a)



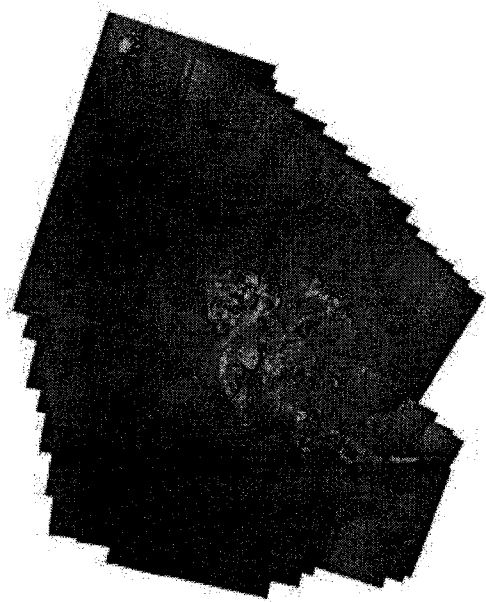
(b)



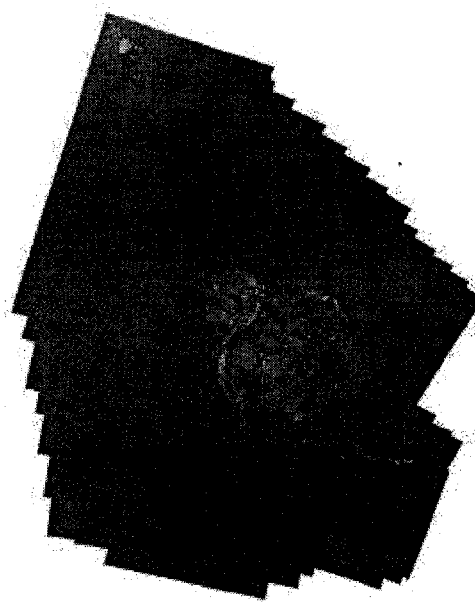
(c)



(d)



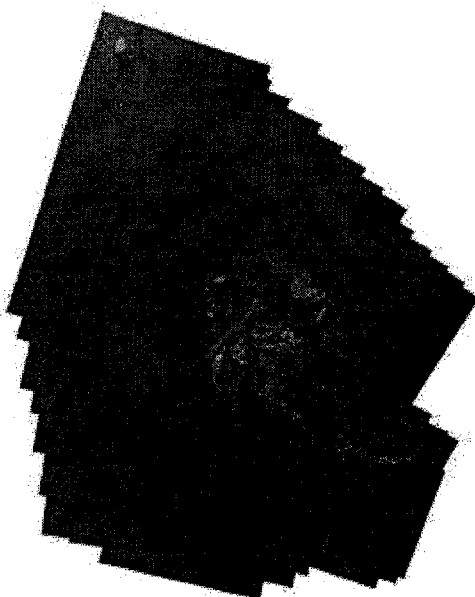
(e)



(f)



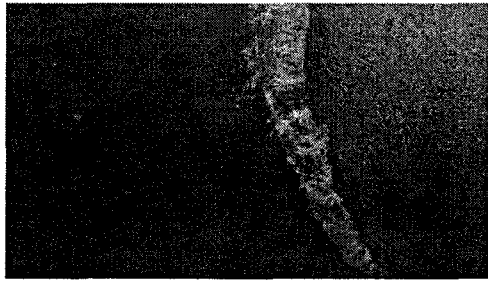
(g)



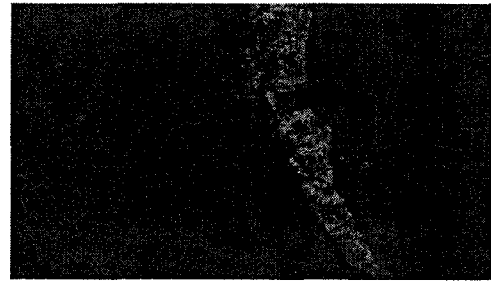
(h)

3m

Figure 4-2 Mosaics of 120 frames using methods of averaging (a)-(b), median (c)-(d), closest patch (e)-(f), and graph-cut (g)-(h), without (left column) and with (right column) illumination correction.



(a)



(b)



(c)



(d)



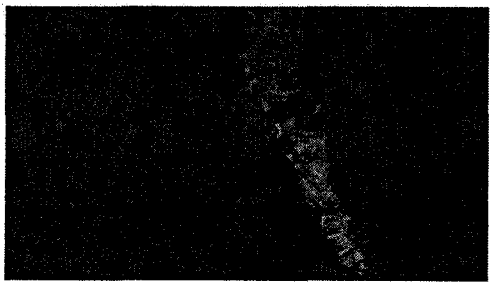
(e)



(f)



(g)



(h)

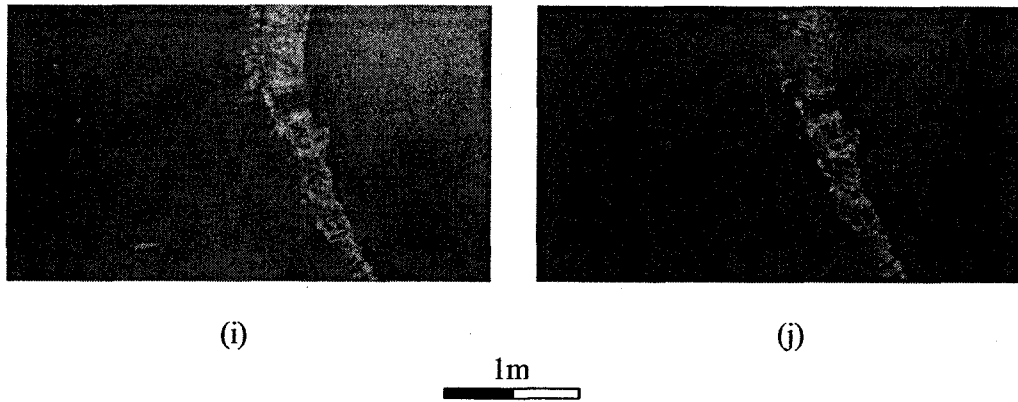


Figure 4-3 Back-projected images (frame 5) from the mosaics using different blending methods. Images without (left column) and with (right column) illumination correction, (a) and (b) are original images content and used as the reference images. (c) and (d) are back-projected images from the averaging mosaics. (e) and (f) are back-projected images from the median mosaics. (g) and (h) are back-projected images from closest patch mosaics. (i) and (j) are back-projected images from graph-cut mosaics.

Comparing the mosaics with and without illumination correction per method, it can be seen that the after the mosaics with illumination correction are more homogeneous and the seams are reduced dramatically. In addition, features in the back-projected images are closer to the original images.

4.2 Perspective Warping and Thin-plate Spline Warping

The perspective warping and thin-plate warping techniques are applied to the illumination corrected images. Feathering (weighted averaging), and graph-cut methods are used as blending methods, and each mosaic is composed of two sequential frames. The pseudocode used to implement thesis techniques given in APPENDIX B.2 for perspective warping, and APPENDIX B.3 for thin-plate spline warping. One group of sample results is shown in Figure 4-4. Another four groups of sample results are given In APPENDIX A.2.

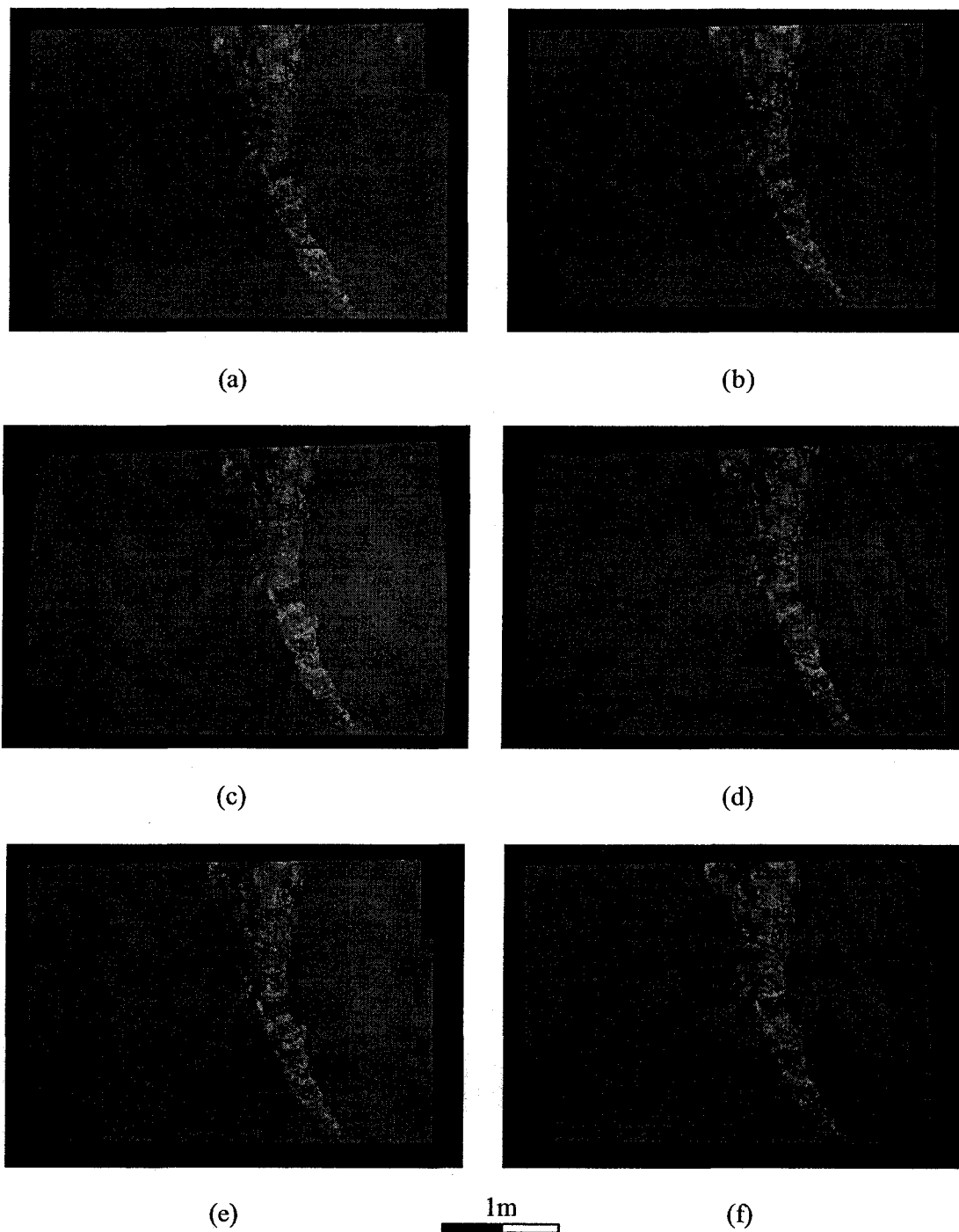


Figure 4-4 Comparisons of warping results on one pair of sequential frames (frame 5 and frame 6) with illumination correction. (a) and (b) are from the images without any warping, (c) and (d) are from perspective warping, and (e) and (f) are from thin-plate spline warping. Mosaics in the left column use the feathering method to blend and mosaics in the right column use the graph-cut method to blend.

The left column of Figure 4-4 shows that blending with the feathering method, after warping, yields clearer features. In addition, the result of thin-plate spline warping is better than perspective warping, which means that the mosaic is improved by having more preserved features in the overlapping region of the mosaic. In the right of Figure 4-4 column, which represents the results for the graph-cut method, a closer observation of the mosaic shows that some features which are lost in the mosaics without warping are kept in the warped ones.

4.3 Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain

The methods proposed are trying to achieve both apparent homogeneity of the illumination and feature preservation. The graph-cut in gradient domain method takes advantage of the graph-cut method which helps preserve features, and the gradient domain technique which helps to improve the homogeneity. A further step of wavelet domain filtering is tried to use the multi-resolution property of the wavelet methods. The pseudocode implemented for these operations is given in APPENDIX B.4 and APPENDIX B.5. One group of sample results is shown in Figure 4-5. Results of eight groups are given in APPENDIX A.3.

	Feathering	Graph-cut
Spatial domain		
Gradient domain		X
Wavelet domain		X

Table 4-1 Comparison of blending methods. The “X” in the table means that the methods are proposed in this thesis, and the rest are shown for comparison.

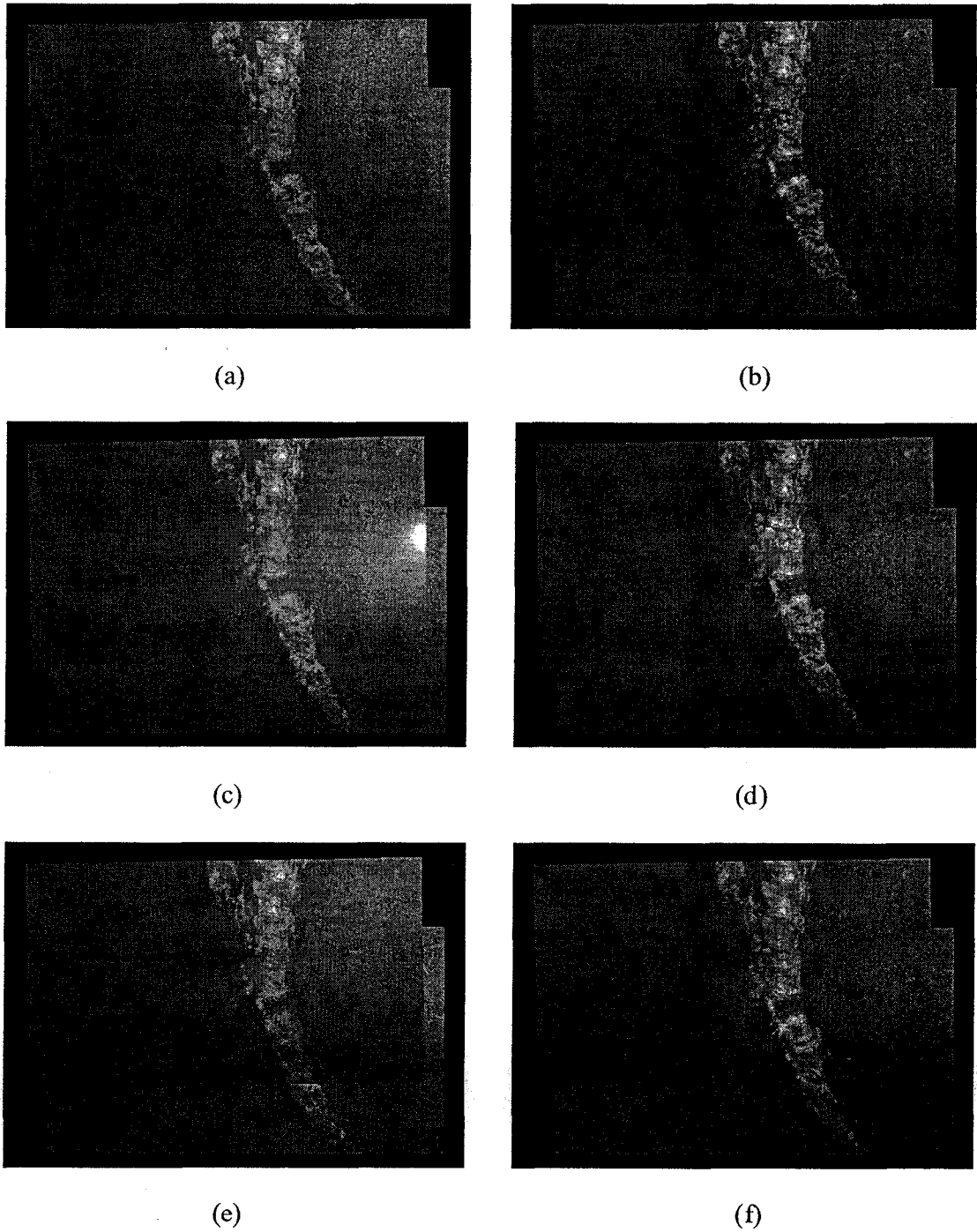


Figure 4-5 Results of mosaics composed of two sequential frames. (a), (c), (e) are blended using feathering in the spatial domain, the gradient domain, and the wavelet domain; and (b), (d), (f) are blended using the graph-cut method in the spatial domain, the gradient domain, and the wavelet domain.

Figure 4-5 shows that the feathering methods are causing blurring and doubling whereas the graph-cut methods do not suffer from these problems. But when the illumination of the frames is different, the mosaic usually has an undesired seam (in spatial graph-cut), which may be mistakenly detected as a feature. The seams can be less pronounced when graph-cut is done in the gradient domain, while in this case, the wavelet domain graph-cut changed the shape of the seam but did not reduce it.

CHAPTER 5

EVALUATION

5.1 Subjective Evaluation

Subjective Evaluation has been widely used to test the quality of visualization methods. In this thesis, subjective evaluation is used to evaluate two groups of methods: 1) perspective warping and thin-plate spline warping, and 2) graph-cut in the gradient domain and graph-cut in the wavelet domain.

5.1.1 Perspective Warping and Thin-plate Spline Warping

Procedure

Five pairs of representative frames and their next sequential frames are selected for the test (APPENDIX A.2). Three different warping methods (non-warping, perspective warping and thin-plate spline warping) are executed on each pair, and for each warping method, both feathering and graph-cut blending methods are applied. This yields the six conditions in Table 5-1.

	Group 1 Feathering	Group 2 Graph-cut
Non-warping	5	5
Perspective Warping	5	5
Thin-plate Spline warping	5	5

Table 5-1 Components of the test set for perspective warping and thin-plate spline warping.

Subjects

Five subjects (all graduate students) with various levels of expertise about mosaicing techniques are selected.

Method

The subjects are given 10 trials for the test. On each trial, subjects are shown a single group of 3 images according to the conditions (from either Group 1 or Group 2). They are asked to select the best one (in terms of the clearness of features) in each trial.

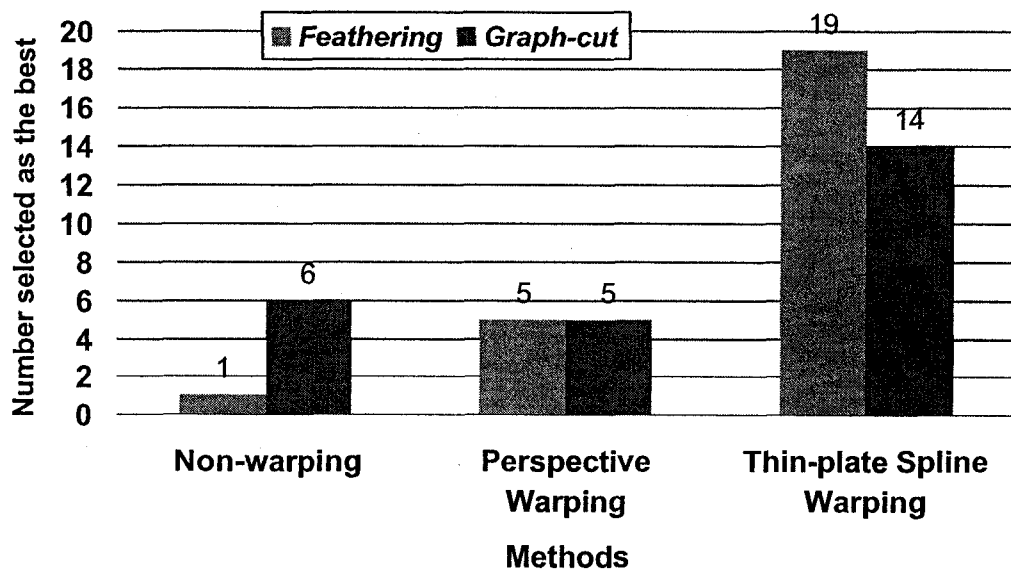


Figure 5-1 Subjective result of perspective warping and thin-plate spline warping.

Figure 5-1 shows the total number of images in each condition selected as best. It suggests that the thin-plate warping is usually deemed better than the perspective warping and non-warping. For example, thin-plate spline warping combined with feathering was chosen as best a total of 19 times, and non-warping combined with feathering was chosen only once.

5.1.2 Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain

Procedure

Eight pairs of representative frames are selected (APPENDIX A.3). For each pair, different blending methods are executed as shown in Table 5-2. In all, there are 6 conditions.

	Feathering	Graph-cut
Spatial domain	8	8
Gradient domain	8	8
Wavelet domain	8	8

Table 5-2 Components of the test set for graph-cut in gradient domain and graph-cut in wavelet domain.

Subjects

Ten subjects (all graduate students) with various levels of expertise with mosaicing techniques are selected.

Method

The subjects are required to choose the overall best mosaic (in terms of homogeneity and clearness of the features) in each subgroup.

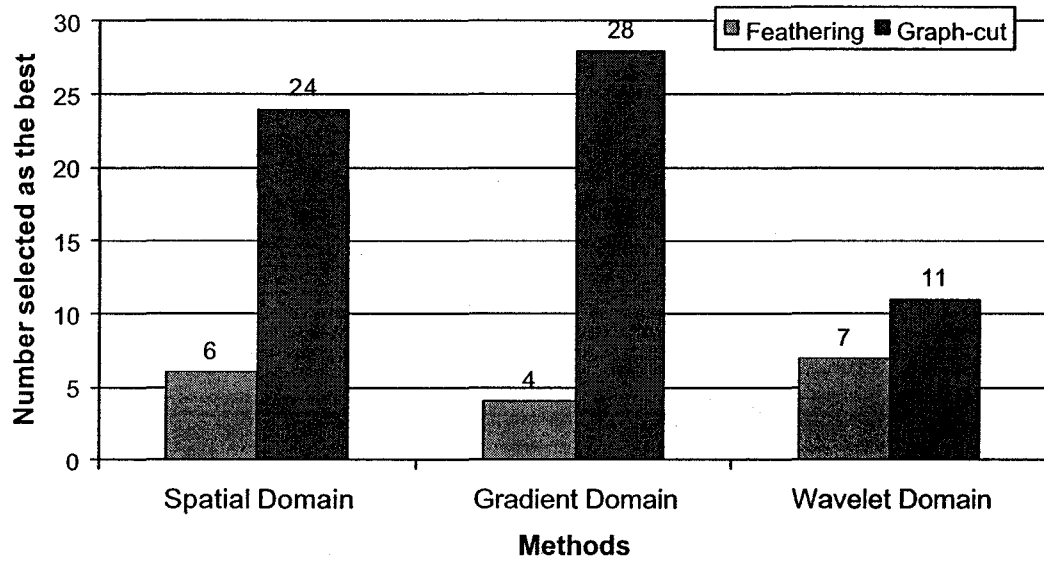


Figure 5-2 Subjective result of feathering, graph-cut in spatial, gradient, and wavelet domains.

The evaluation in Figure 5-2 suggests that the graph-cut method is usually deemed better than the feathering method for the set of images chosen in these three domains. The graph-cut in gradient domain method is usually deemed the best in the overall performance.

5.2 Objective Evaluation

Criteria of objective evaluation for mosaicing are seldom considered in the literature. Mostly, researchers subjectively comment on the mosaics' appearance. In this section, objective methods for the purpose of quality or feature evaluations are tested including the standard image quality evaluation method of Peak Signal to Noise Ratio (PSNR), the Universal Image Quality Index (UIQI) [WB02], and the edge based objective evaluation method [PX05]. In this thesis, subjective evaluation is considered to

be reliable [PX05], and the objective evaluation is compared to the subjective evaluation in order to test its validity.

5.2.1 Graph-cut in Gradient Domain and Graph-cut in Wavelet Domain

Objective evaluation in this thesis is used for methods including: graph-cut in the gradient domain and graph-cut in the wavelet domain, both of which can be formulated using existing evaluation methods.

The PSNR and UIGI formulas are Equation 2-47 and 2-54, respectively, and the inputs of the equations are the original image and back-projected image from the mosaics. Because the test mosaic is composed of two sequential images, the PSNR or UIGI is obtained from each of the two images respectively and the average value is calculated as the final PSNR or UIGI of that test mosaic. The same set of data from the corresponding subjective evaluation is used in these objective evaluation methods. For each method, the average value of PSNR or UIGI is calculated from the eight mosaics.

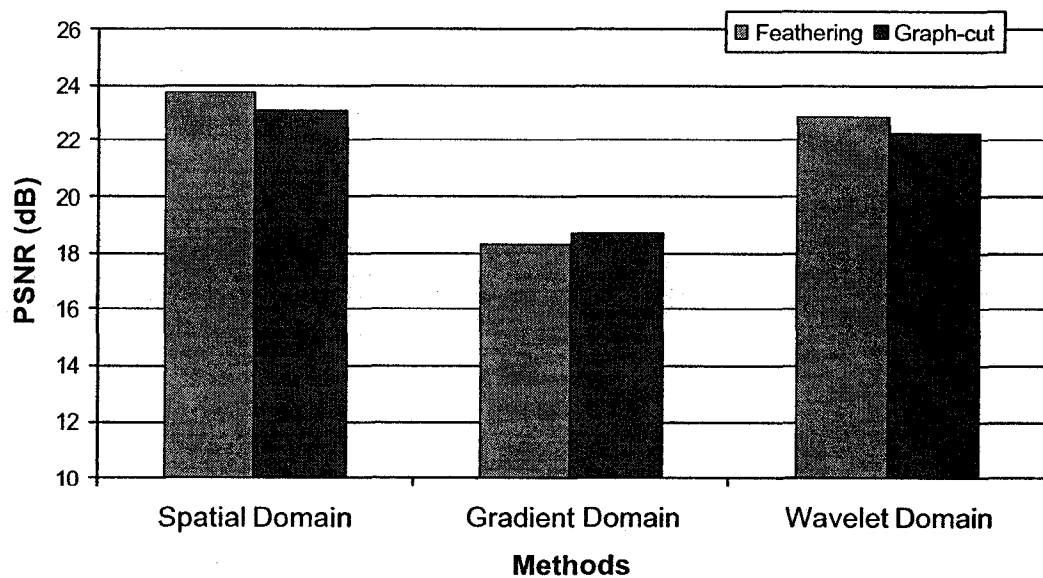


Figure 5-3 Average PSNR values of the blending methods.

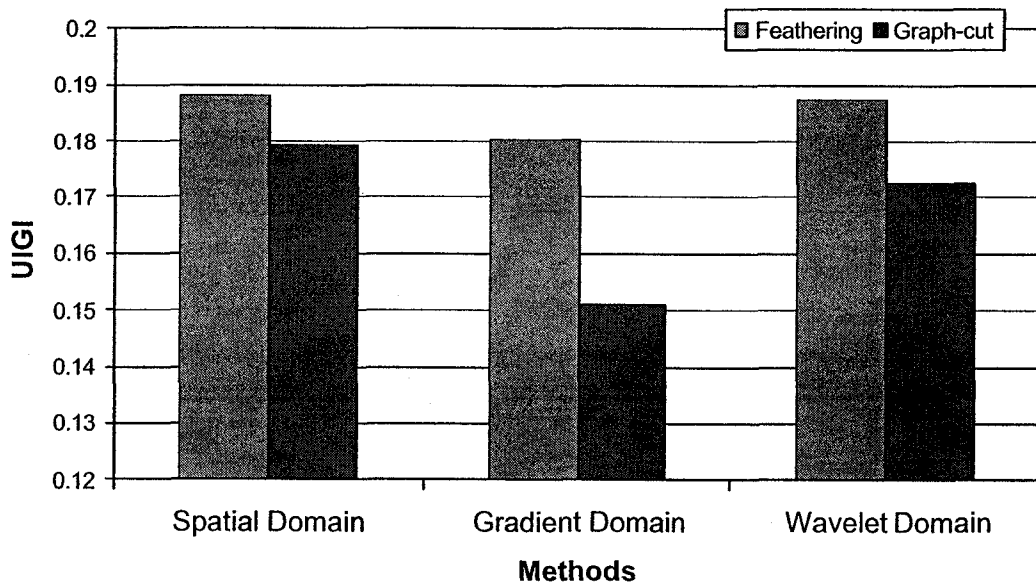


Figure 5-4 Average UIGI values of the blending methods.

According to both the PSNR test (Figure 5-3) and UIGI test (Figure 5-4), the graph-cut in gradient domain method is one of the least effective.

However, this result is not consistent with the subjective evaluation (Figure 5-2) which showed that graph-cut in the gradient domain is deemed to be the best. This discrepancy may be due to the fact that the graph-cut in the gradient domain method produces larger changes in the absolute gray level values.

Edge based objective evaluation (Equation 2-59) is also tested on the same data set, and similarly, the average value of $Q^{AB/F}$ is calculated from the eight mosaics.

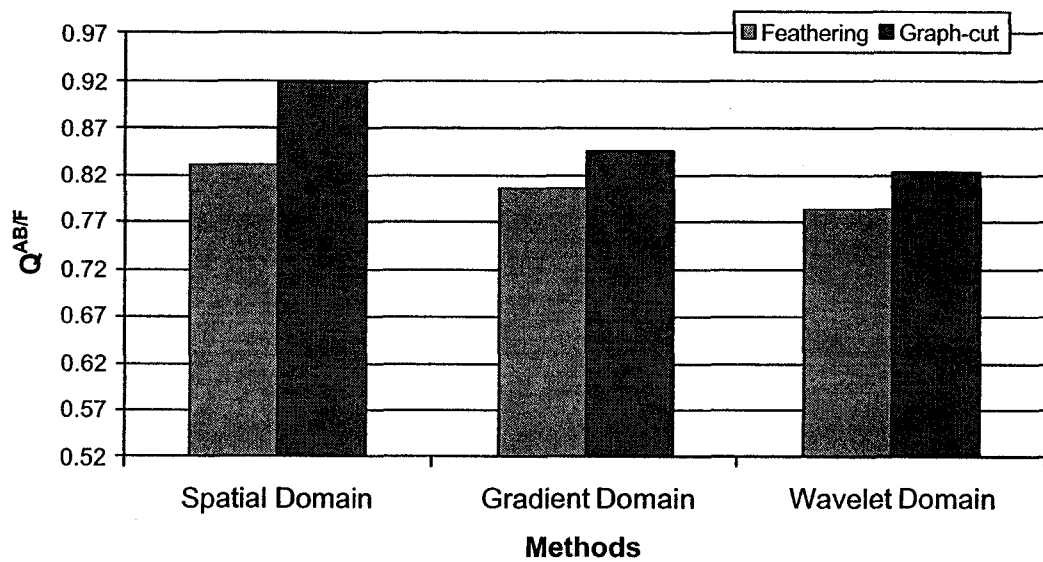


Figure 5-5 Edge based objective evaluation results.

In the edge based objective evaluations (Figure 5-5), the graph-cut method results are higher than the feathering results, which suggests that graph-cut methods have a statistically better performance in feature preservation. This is consistent with the subjective evaluation (Figure 5-2), so the edge based objective evaluation is helpful in measuring feature preservation.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

In this thesis, several methods to improve blending techniques for underwater mosaics are proposed, implemented, evaluated, and the results are compared. These methods help to obtain an image having the appearance of homogeneous illumination as well as preserving features. They take advantage of techniques used in image processing, computer vision, and computer graphics, and combine them for mosaicing underwater images.

The purposes of blending are: First, to obtain a consistent homogeneous mosaic which is seamless. In achieving this goal, the thesis explored the techniques of detrending, the gradient domain techniques, and wavelet domain techniques. Second is to preserve the features, which means to avoid doubling and blurring problems, and the graph-cut and warping were evaluated.

The median mosaic based illumination correction method is proposed to relight the original frame according to the back-projected images from median mosaics. The histograms show that the pixel ranges are shortened after correction, which means that the images are more homogeneous. The seams are diminished after illumination correction in the mosaics of 120 frames and the single back-projected images shown in the results.

In order to preserve the features, perspective warping and thin-plate warping are tested. It is known that the perspective distortion is caused by the uncontrolled camera motion. Perspective warping is proposed to reduce this distortion. The experiments show that although this “perspective” approach can lead to some improvement for two of these overlapping frames (gray value differences are reduced, and the mosaic appears more consistent visually), the examples with many (over 5) overlapping frames almost always converged to a state with insignificantly reduced gray value differences and most importantly, the method fails to match features in overlapping images. This result can be interpreted as indirect evidence that for scenes with 3D content the use of perspective model as an alternative to rigid affine models does not improve the mosaicing result.

Another technique for warping used in this thesis is the thin-plate spline warping, which chooses control points according to the content of the images. The control points are the interest points extracted by feature detection algorithms. The advantage of thin-plate spline warping over perspective warping is that warping occurs locally which reduces distortions in the regions which do not need to warp. This type of warping is tested on pairs of frames, and the results show that doubling and blurring are significantly reduced. The subjective results also show that thin-plate spline warping has a better performance than perspective warping.

Two other methods for obtaining homogeneous mosaics with preserved features were proposed, implemented and evaluated, namely graph-cut in the gradient domain and graph-cut in the wavelet domain. From the results, it can be observed that the graph-cut in the gradient domain method helps achieve homogeneity of the mosaic, and preserve the features. However, these methods are not practical to use for more than two images

because of error accumulations. An unsolved problem in the experiment is that in some cases, the reconstructions are not stable. This may be due to misalignment of the images in the gradient domain recovery calculations. In the experiments of graph-cut in the wavelet domain, Haar wavelet, which is simple to implement, is used, and the graph-cut is performed on the first order wavelet coefficients. Observations of the results show that this method only changed the seam shape and made the seams somewhat blurred. Overall, this method was less effective than graph-cut in the gradient domain.

Subjective evaluation of the results shows that the graph-cut in the gradient domain method is better at achieving the appearance of homogeneous illumination and preservation of features. Objective quality measurement methods are also used and the results are compared to the subjective evaluation. The PSNR and UIGI methods show that the image quality is lower using the method of graph-cut in the gradient domain. This conflicts with the subjective results, therefore these two objective evaluation methods are not informative in evaluating the blending methods proposed in this thesis.

Edge based objective evaluation which measures the feature preservation property of the blending methods is also tested against subjective evaluations. The objective results show that the graph-cut methods have a better performance than the feathering methods, which is consistent with the subjective evaluation, therefore the edge based objective evaluation is informative in measuring the feature preservation in this thesis.

In summary, from the experimental results and analysis presented in this thesis, the combination of median mosaic based illumination correction and thin-plate spline warping have more potential in achieving apparent homogeneity of illumination and feature preservation in mosaics. The graph-cut method is recommended for image

mosaicing. The future work lies in the application of these methods to large mosaics and accelerating the computation speed. In the method of thin-plate spline warping, more robust methods for finding interest point to be used as control point are also needed.

REFERENCES

- [AC00] A. Almansa, L. Cohen, "Fingerprint Image Matching by Minimization of A Thin-Plate Energy using A Two-step Algorithm with Auxiliary Variables", Proc. of the IEEE Workshop on Applications of Computer Vision, pp. 53-40, Dec. 2000.
- [ADADCCSC04] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colbrn, B. Curless, D. Salesin, and M. Cohen, "Interactive Digital Photomontage". ACM Transactions on Graphics 23(3), pp. 294-302, Aug. 2004.
- [B89] F. L. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations", IEEE Transaction of Pattern Analysis and Machine Intelligence, vol. 11, No. 6, pp. 567-585, Jun. 1989.
- [BA83] P. J. Burt, and E. H. Adelson, "A Multiresolution Spline with Application to Image Mosaics", ACM Transactions on Graphics 2(4), pp. 217-236, Oct. 1983.
- [BCVBS01] C. Ballester, V. Caselles, J. Verdera, M. Bertalmio, and G. Sapiro, "Filling-in by Joint Interpolation of Vector Fields and Gray Levels", IEEE Transactions on Image Processing and Signal Analysis, pp. 1200-1211, 2001.
- [BSMYMWSP02] R. D. Ballard, L. E. Stager, D. Master, D. R. Yoerger, D. A. Mindell, L. L. Whitcomb, H. Singh, and D. Piechota, "Iron Age Shipwrecks in Deep Water off Ashkelon, Israel", American Journal of Archaeology, 106(2): 151-168, Apr. 2002.
- [BVZ01] Y. Boykov, O. Veksler, and R. Zabih. "Fast Approximate Energy Minimization via Graph-cuts", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222--1239, Nov. 2001.

[DL01] I. Drori, D. Lischinski, "Fast Multi-Resolution Image Operations in the Wavelet Domain", IEEE Transactions on Visualization and Computer Graphics, pp. 395-411, 2003.

[EF01] A. Efros and W. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH2001, Proceedings of Computer Vision, pp. 1033-1038, 2001.

[ESH00] R. Eustice, H. Singh and J. Howland, "Image Registration Underwater for Fluid Flow Measurements and Mosaicking", Proc. of the MTS/IEEE OCEANS conference, vol. 3, pp. 1529-1534, 2000.

[FC88] R.T. Fankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms", IEEE Transaction on Patten Analysis and Machine Intelligence, 10(4): 439-451, 1988.

[FLW02] R. Fattal, D. Lischinski, and M Werman, "Gradient Domain High Dynamic Range Compression", Proc. of ACM SIGGRAPH 2002, Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH, pp. 249-256, Jul. 2002

[W96] A. J. Williamson, "Quantum Monte Carlo Calculations of Electronic Excitations", <http://www.tcm.phy.cam.ac.uk/~ajw29/thesis/node15.html>, 1996

[FB81] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381—395, 1981.

[FY06] F. Gu, Yuri Rzhanov, "Optical Image Blending for Underwater Mosaics", IEEE Oceans' 2006, Boston, MA, pp.1-5, Sep.2006.

[GGNM06] N. Gracias, A. Gleason, S. Negahdaripour, and M. Mahoor, "Fast Image Blending using Watersheds and Graph Cuts", pp.1-10, BMVC 2006.

[GNC02] R. Garcia, T. Nicosevici, X. Cufi. "On the Way to Solve Lighting Problems in Underwater Imaging", Oceans'02 MTS/IEEE, pp. 1018-1024, 2002.

[GZBV03] N. R. Gracias, S. van der Zwaan, A. Bernardino, and J. S. Victor, "Mosaic-Based Navigation for Autonomous Underwater Vehicles", IEEE Journal of Oceanic Engineering, vol. 28, no. 4, pp. 609-624, Oct. 2003.

[KJH02] D.-K. Kim, B.-T. Jang, C.-J. Hwang, "A Planar Perspective Image Matching using Point Correspondences and Rectangle-to-Quadrilateral Mapping", Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 87-91, 2002.

[KSE03] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts", ACM Transactions on Graphics, 22(3):277--286, Jul. 2003.

[L04] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, pp. 91-110, 2004.

[LZPW04] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless Image Stitching in the Gradient Domain", In Eighth European Conference on Computer Vision, vol. 4, pp. 377-389, 2004.

[SA90] E. P. Simoncelli and E. H. Adelson, "Non-separable Extensions of Quadrature Mirror Filters to Multiple Dimensions", Proc. of the IEEE Special Issue on Multi-dimensional Signal Processing 78(4), pp. 652-664, Apr. 1990.

[MYF03] H. Moriyama, N. Yamashita, and M. Fukushima, "The Incremental Gauss-Newton Algorithm with Adaptive Step-size Rule", *Computational Optimization and Applications*, vol 26, pp. 107 -141, 2003.

[OSS68] A. V. Oppenheim, R. W. Schaffer and T. G. Stockham Jr., "Non-linear Filtering of Multiplied and Convolved Signals". *Proc. IEEE*, vol. 56, no. 8, pp. 1264-1291, 1968.

[PD84] T. Porter and T. Duff "Compositing Digital Images", *Computer Graphics* vol. 18, no. 3 pp 253-259, Jul. 1984.

[PGB03] P. Perez, M. Gangnet, and A. Blake. "Poisson Image Editing", *Proceedings of SIGGRAPH 2003*, pp. 313--318, Jul. 2003.

[PTVWF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing", 2nd ed. Cambridge University Press, 1992.

[PX05] V. Petrovic, and C. Xydeas, "Objective Evaluation of Signal-level Image Fusion Performance", *Optical Engineering*, vol. 44, 087003: 1-8, Aug. 2005.

[QY05] X. Qin, and Y.-H. Yang, "Theoretical Analysis of Graphcut Textures", <http://www.cs.ualberta.ca/TechReports/2005/TR05-26/TR05-26.pdf>

[RLF00] Y. Rzhanov, L. Linnett, and R. Forbes "Underwater Video Mosaicing for Seabed Mapping", *Proc. of the IEEE Conf. on Image Processing*, vol. 1, pp. 224-227, 2000.

[S05] J. A. Snyman. "Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms", Springer Publishing, 2005.

[SERPACT04] H. Singh, R. Eustice, C. Roman, O. Pizarro, R. Armstrong, F. Gilbes, and J. Torres, "Imaging coral I: Imaging coral habitats with the SeaBED AUV", *Subsurface Sensing Technologies and Applications*, 5(1):25-42, Jan. 2004.

[SHC01] M. Su, W. Hwang, K. Cheng, "A Variational Calculus Approach to Multiresolution Image Mosaic" *Proc. of International Conference on Image Processing*, vol 2, pp: 245-248, Thessaloniki, Greece, 2001.

[SHYW98] H. Singh, J. Howland, D. Yoerger and L. Whitcomb, "Quantitative Photomosaicing of Underwater Imaging", *Proc. MTS/IEEE OCEANS*, vol. 1, pp. 263-266, 1998.

[SS00] H.-Y. Shum and R. Szeliski. "Construction of Panoramic Image Mosaics with Global and Local Alignment", *International Journal of Computer Vision*, 36(2): 101-130, 2000.

[OFT02] F. Odone, A. Fusiello, and E. Trucco. "Layered Representation of A Video Shot with Mosaicing", *Pattern Analysis and Applications*, 5(3): 296-305, Aug. 2002.

[TFTMM02] J. Tregellas, R. Freedman, J. Tanabe, D. Miller and F. G. Meyer. "Comparison of Detrending Methods for Optimal fMRI Preprocessing", *NeuroImage*, pp. 902-907, 2002.

[V99] O. Veksler, "Efficient Graph-based Energy Minimization Methods in Computer Vision", PhD thesis, Department of Computer Science, Cornell University, 1999.

[W01] Y. Weiss. "Deriving Intrinsic Images from Image Sequences", Proc. of International Conference on Computer Vision, pp. 68--75, 2001.

[WB02] Z. Wang and A. C. Bovik, "A Universal Image Quality Index", IEEE Signal Processing Letters, vol. 9. no. 3. pp. 81-84, 2002.

[YBCRW00] D. R. Yoerger, A. M. Bradley, M. H. Cormier, W. B. F. Ryan, and B. B. Walden, "Fine-scale Seafloor Survey in Rugged Deep-ocean Terrain with An Autonomous Robot", IEEE international Conference on Robotics and Automation, vol.2, pp. 1767-1774, San Francisco, USA, 2000.

APPENDICES

APPENDIX A

DATA RESULTS

A.1 Data Results for Median Mosaics Based Illumination Correction

The followings are results for comparison of median mosaics based illumination correction. Left column contains the back-projected images from the mosaic without illumination correction, and the right column contains the back-projected images with illumination correction. (a) and (b) are back-projected images from the averaging mosaics. (c) and (d) are back-projected images from the median mosaics. (e) and (f) are back-projected images from closest patch mosaics. (g) and (h) are back-projected images from graph-cut mosaics. The contents are referring to section 3.1 and section 4.1 in the thesis.

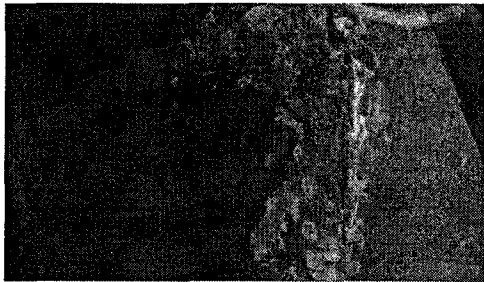
Frame 9



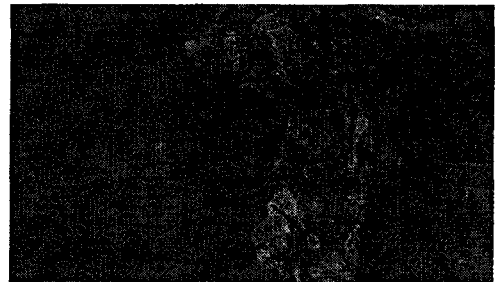
(a)



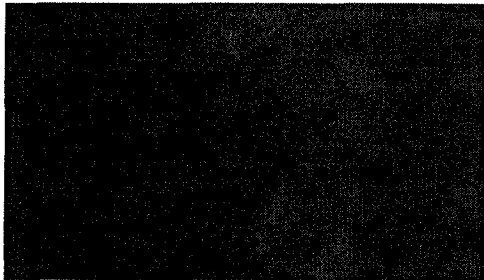
(b)



(c)



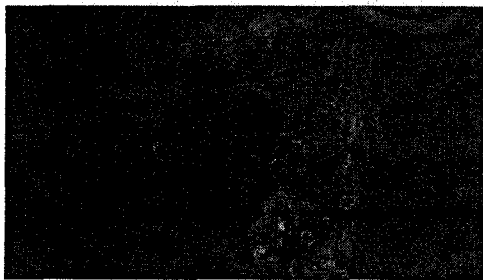
(d)



(e)



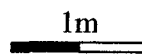
(f)



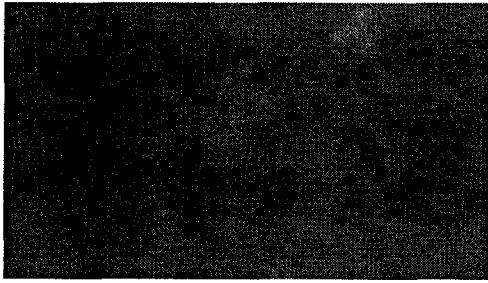
(g)



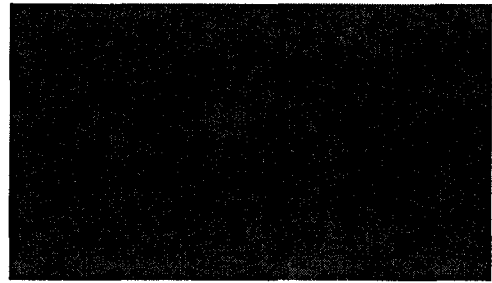
(h)



Frame 10



(a)



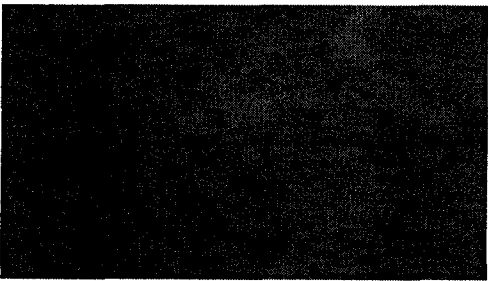
(b)



(c)



(d)



(e)



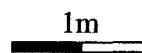
(f)



(g)



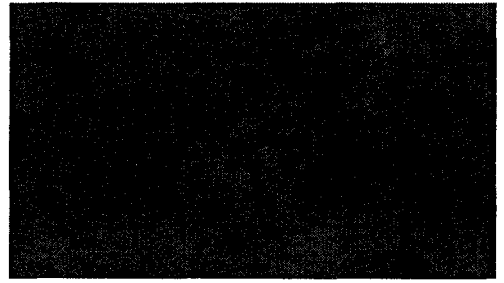
(h)



Frame 11



(a)



(b)



(c)



(d)



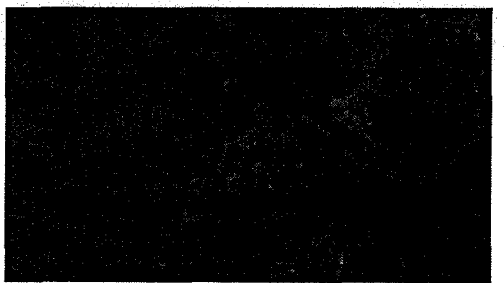
(e)



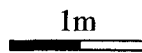
(f)



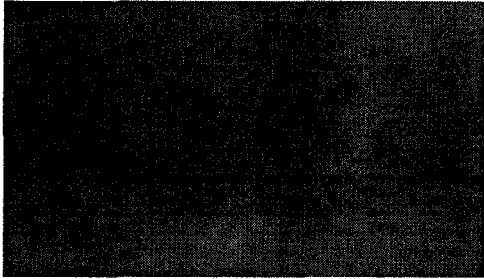
(g)



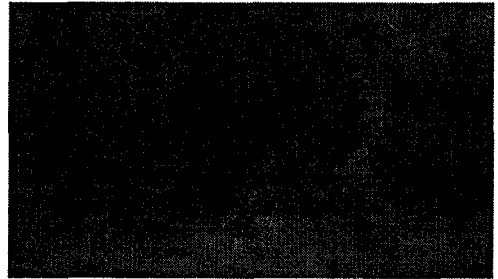
(h)



Frame 12



(a)



(b)



(c)



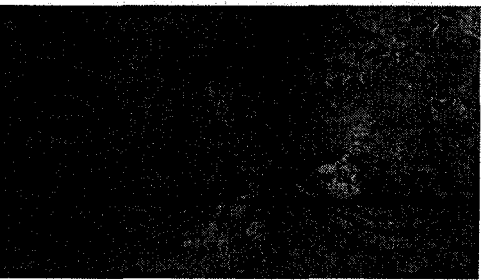
(d)



(e)



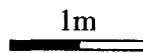
(f)



(g)



(h)



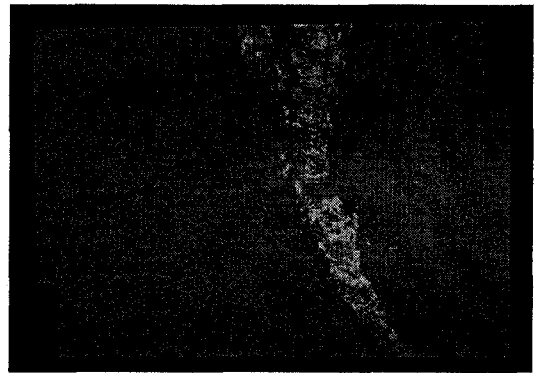
A.2 Data Results for Perspective Warping and Thin-plate Spline Warping

Mosaics composed of two sequential frames after illumination correction. (a) and (b) are from the images without any warping, (c) and (d) are from perspective warping, and (e) and (f) are from thin-plate spline warping. Mosaics in the left column are mosaics using the feathering method and in the right column are using graph-cut methods. The contents are referring to section 3.2, section 3.3 and section 4.2 in the thesis.

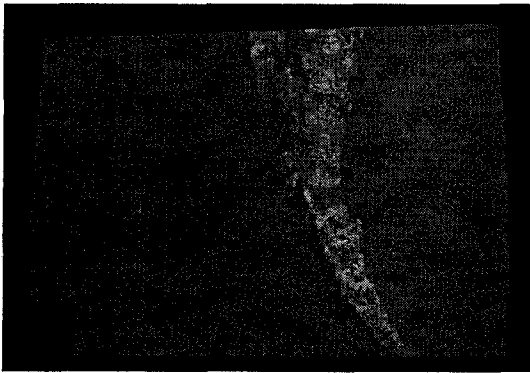
Pair 1 (Frame 5 and Frame 6)



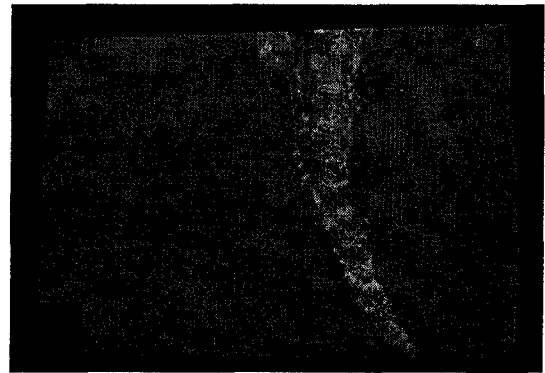
(a)



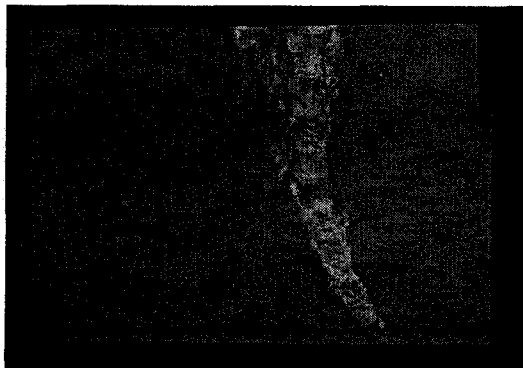
(b)



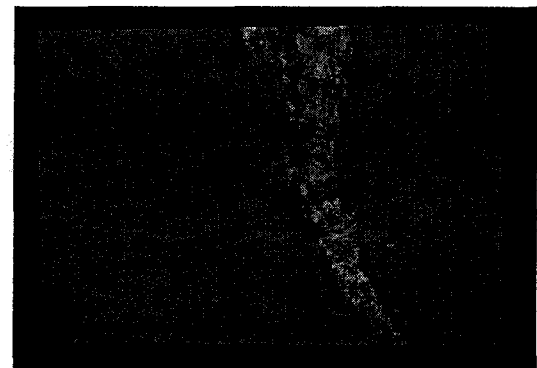
(c)



(d)




(e)

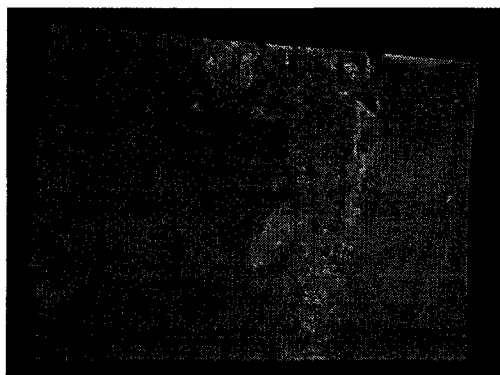


(f)

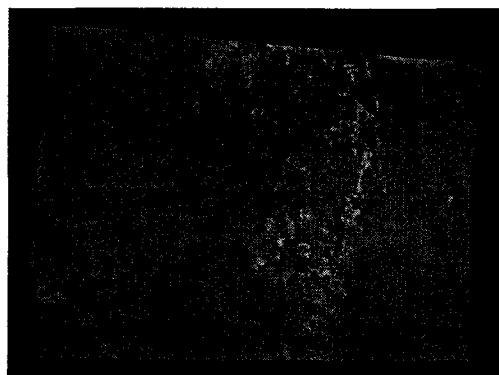
1m

A horizontal scale bar with a black outline and a white fill, positioned below the text '1m'.

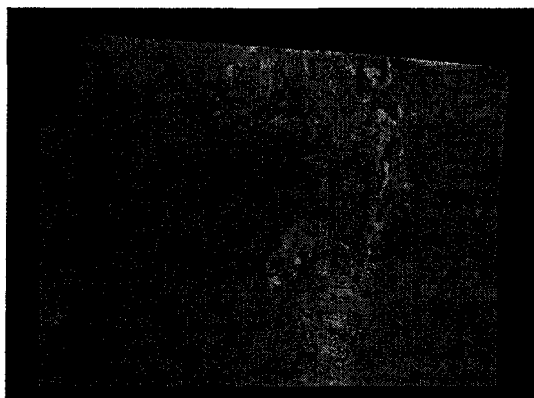
Pair 2 (Frame 9 and Frame 10)



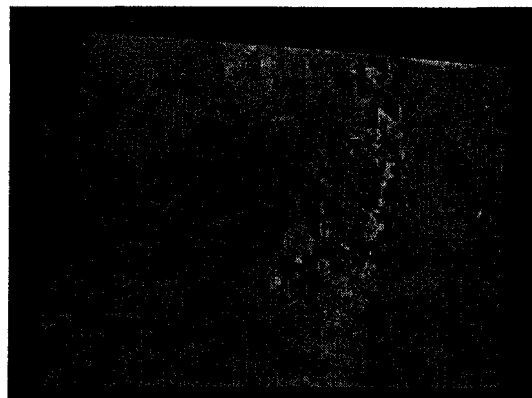
(a)



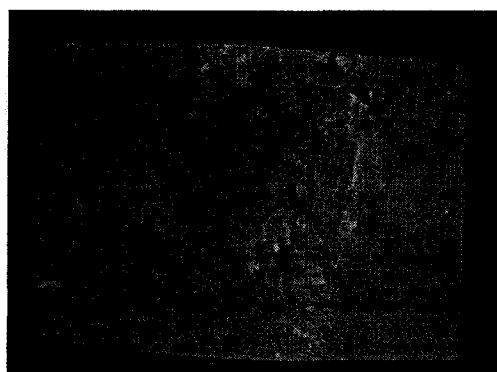
(b)



(c)



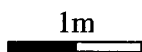
(d)



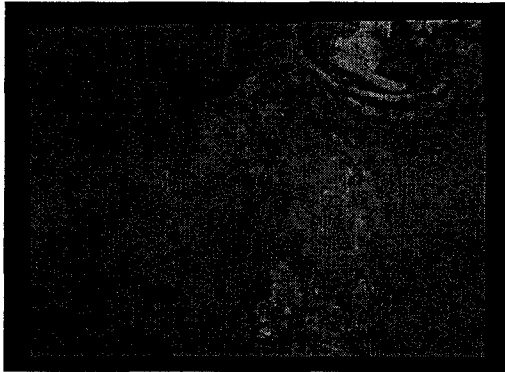
(e)



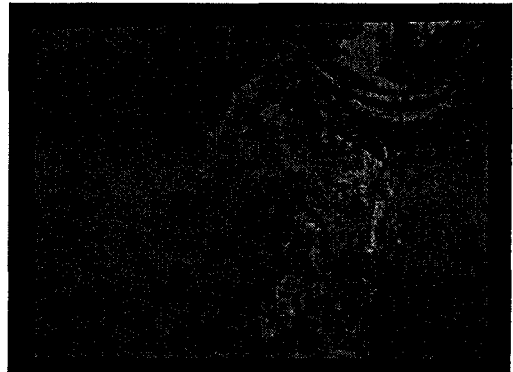
(f)



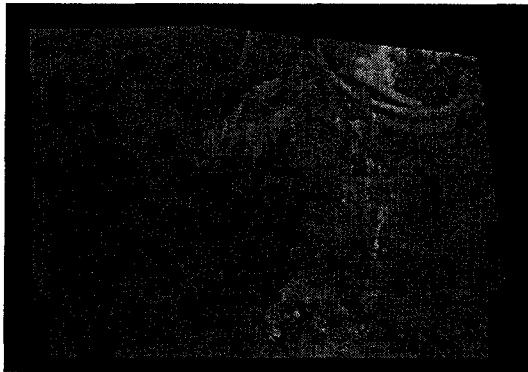
Pair 3 (Frame 13 and Frame 14)



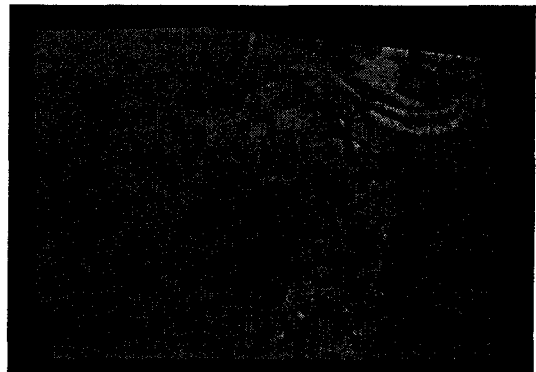
(a)



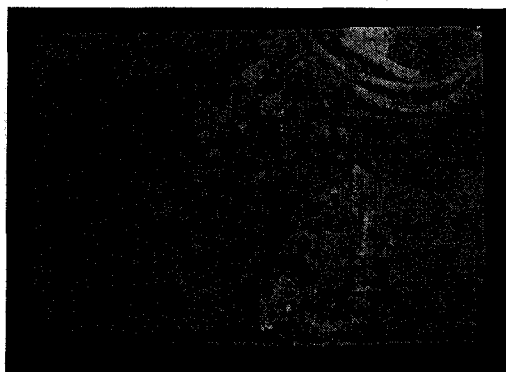
(b)



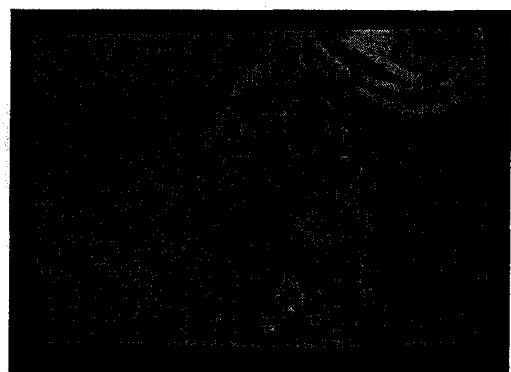
(c)




(d)



(e)



(f)

1m


A horizontal scale bar with a black top half and a white bottom half, used for size reference.

Pair 4 (Frame 8 and Frame 9)



(a)



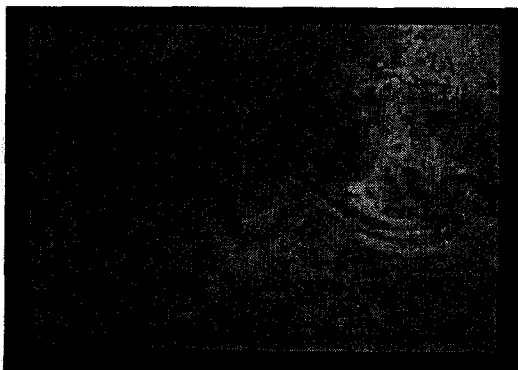
(b)



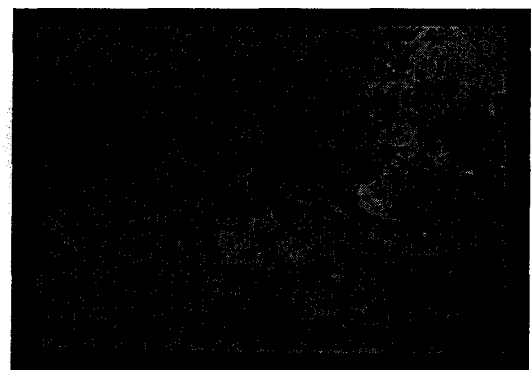
(c)



(d)



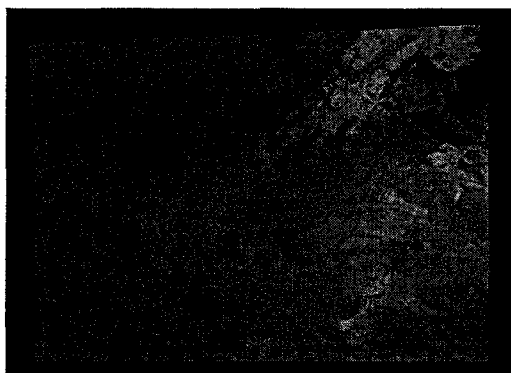
(e)



(f)

1m
—|—

Pair 5 (Frame 11 and Frame 12)



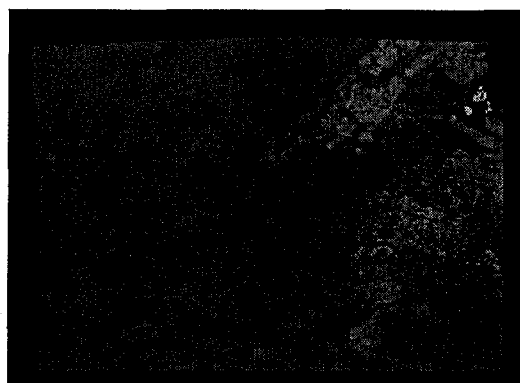
(a)



(b)



(c)



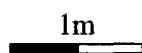
(d)



(e)



(f)



A.3 Data Results for Graph-cut in Gradient Domain and Graph-cut in Wavelet

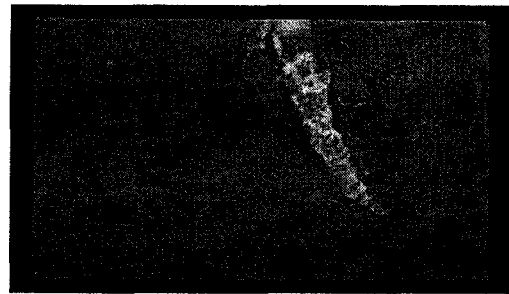
Domain

Mosaics composed of two sequential frames. (a), (c), (e) are blended using feathering in the spatial domain, the gradient domain, and the wavelet domain respectively; and (b), (d), (f) are blended using graph-cut method in spatial domain, the gradient domain, and the wavelet domain. The contents are referring to section 3.4, section 3.5 and section 4.3 in the thesis.

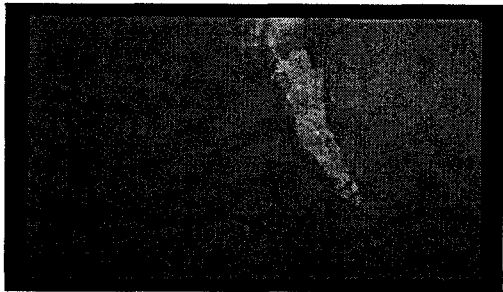
Pair 1 (Frame 3 and Frame 4)



(a)



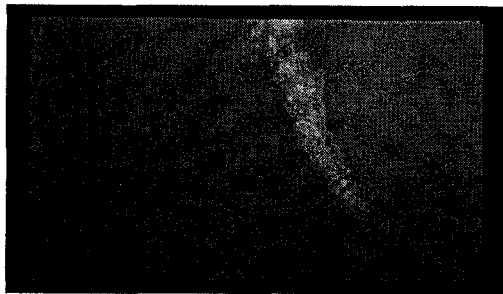
(b)



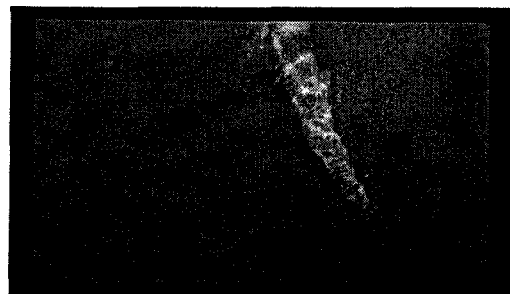
(c)



(d)



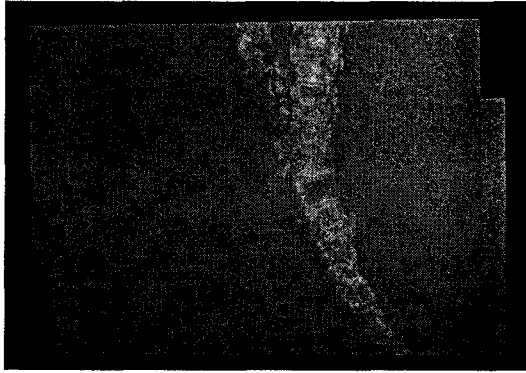
(e)



(f)

1m
—|—

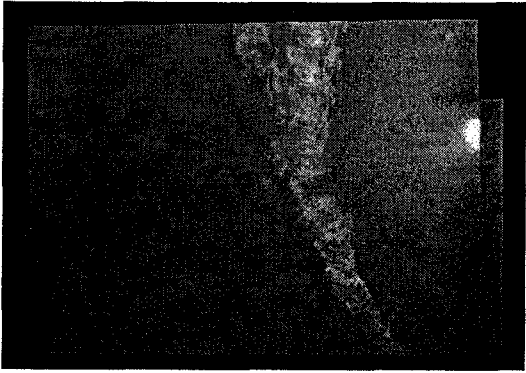
Pair 2 (Frame 5 and Frame 6)



(a)



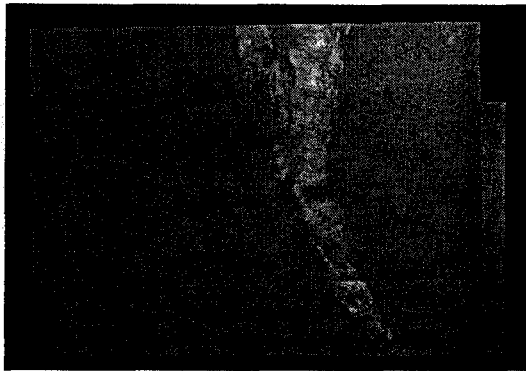
(b)



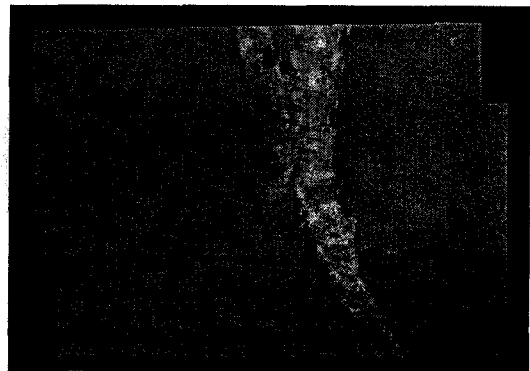
(c)



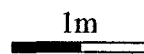
(d)



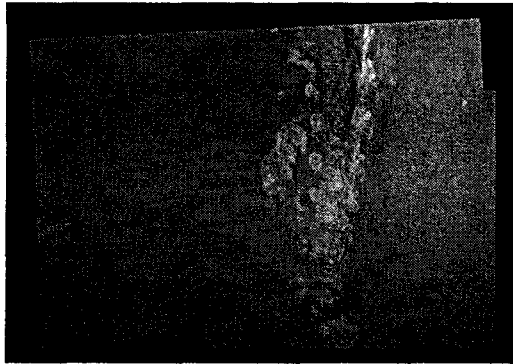
(e)



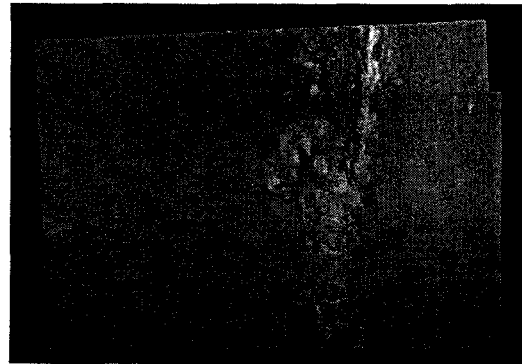
(f)



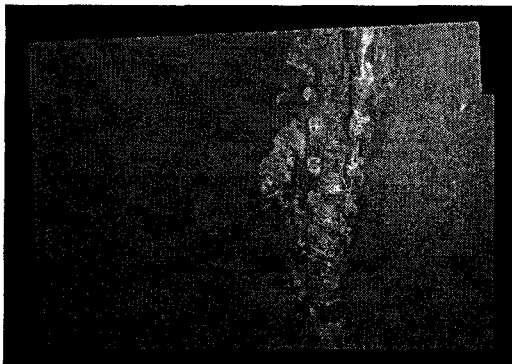
Pair 3 (Frame 7 and Frame 8)



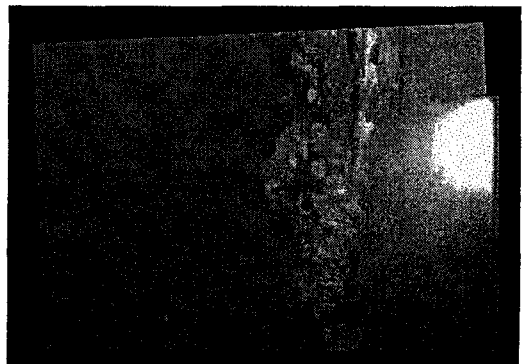
(a)



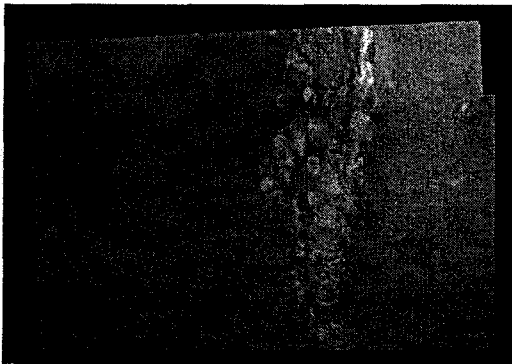
(b)



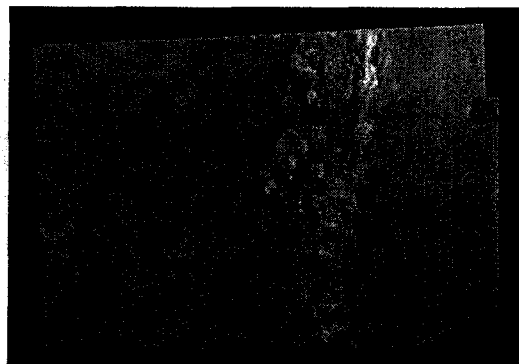
(c)



(d)



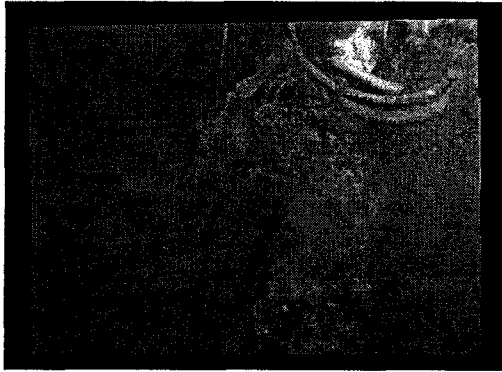
(e)



(f)

1m
—|—

Pair 4 (Frame 9 and Frame 10)



(a)



(b)



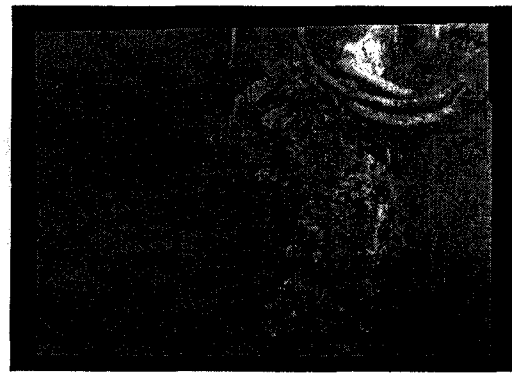
(c)



(d)



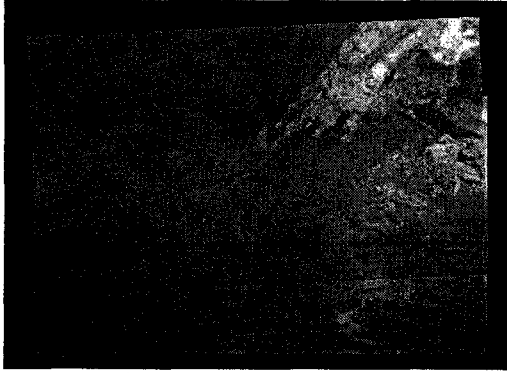
(e)



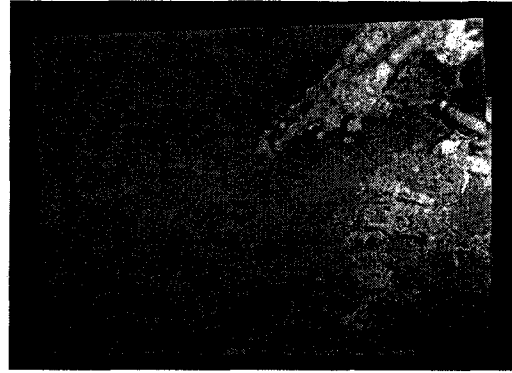
(f)

1m
—|—

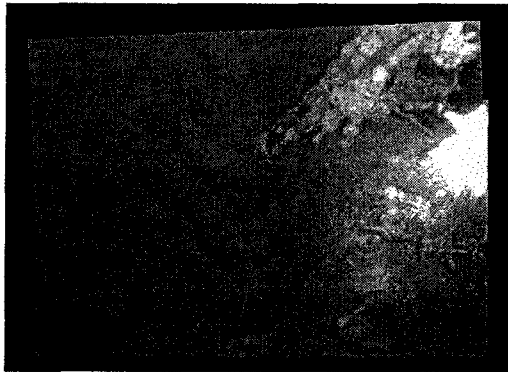
Pair 5 (Frame 11 and Frame 12)



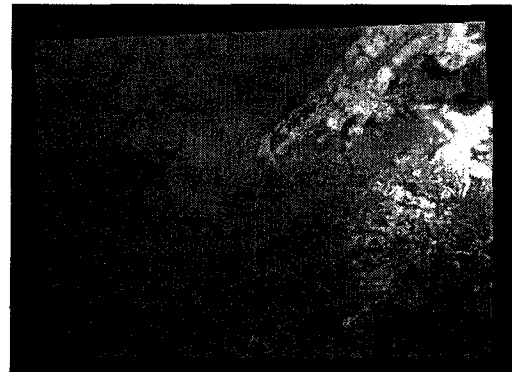
(a)



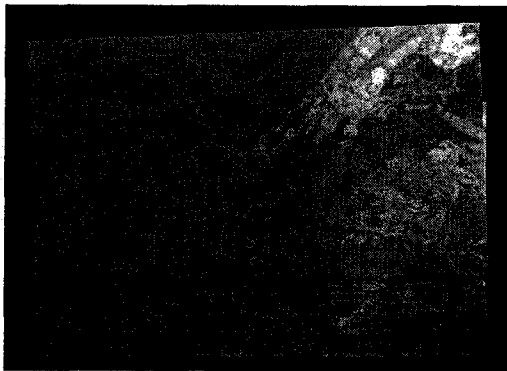
(b)



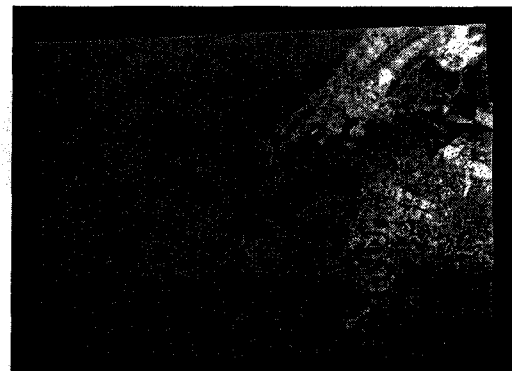
(c)




(d)



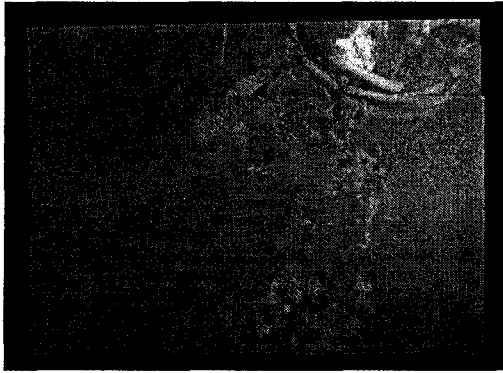
(e)



(f)

1m


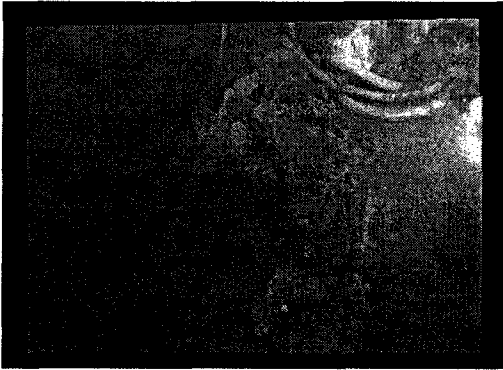
Pair 6 (Frame 13 and Frame 14)



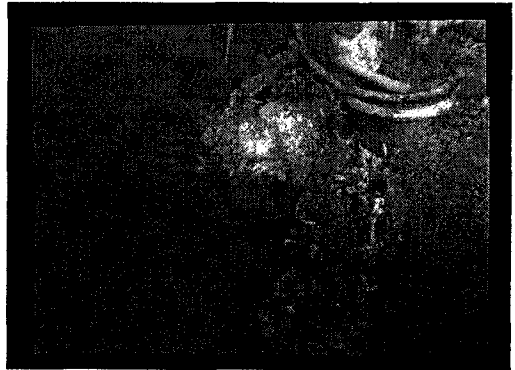
(a)



(b)



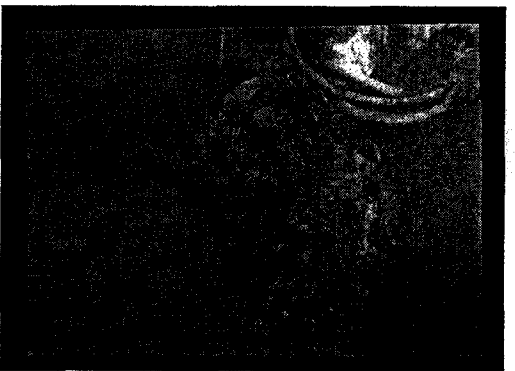
(c)



(d)



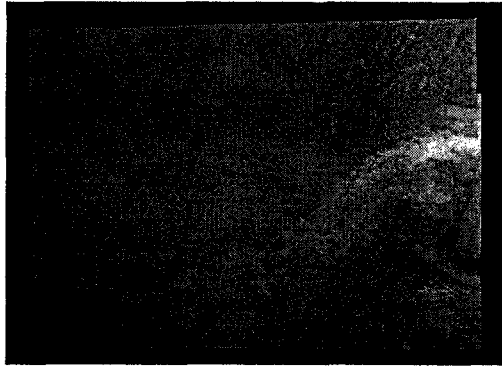
(e)



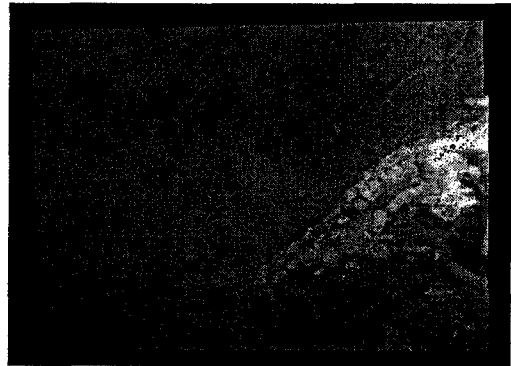
(f)

1m
—

Pair 7 (Frame 15 and Frame 16)



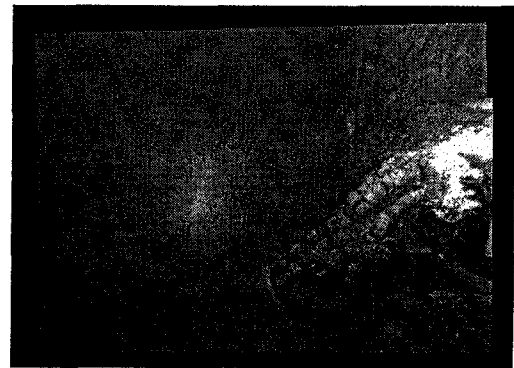
(a)



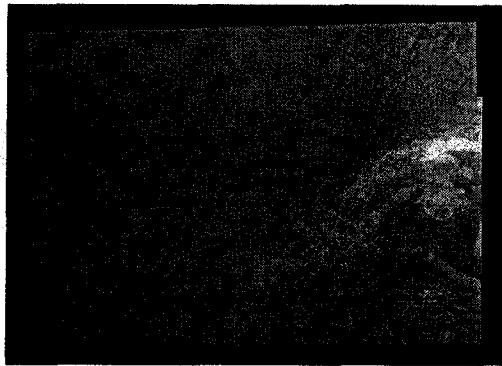
(b)



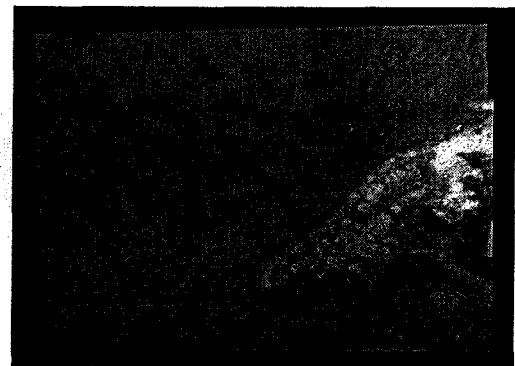
(c)



(d)



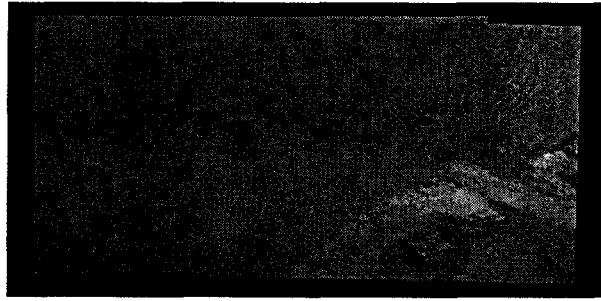
(e)



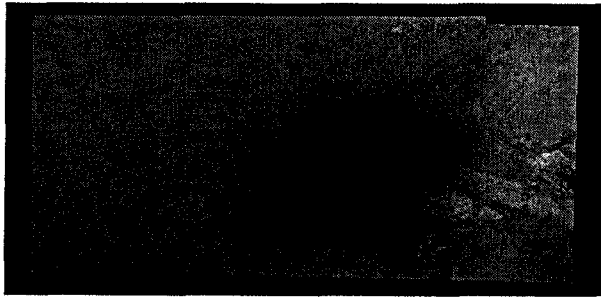
(f)

1m
—|—

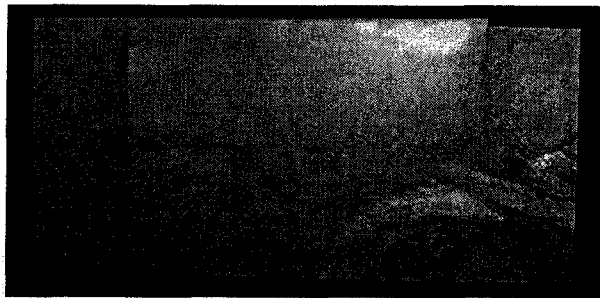
Pair 8 (Frame 17 and Frame 18)




(a)

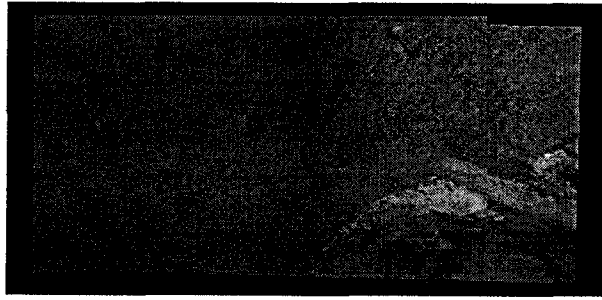


(b)

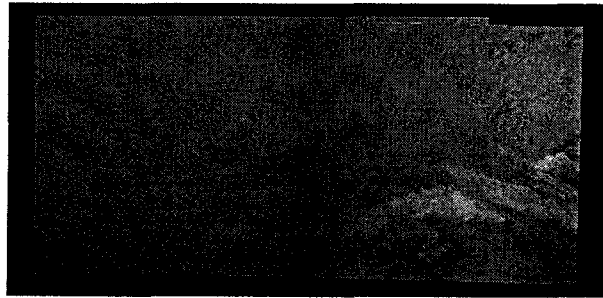


(c)

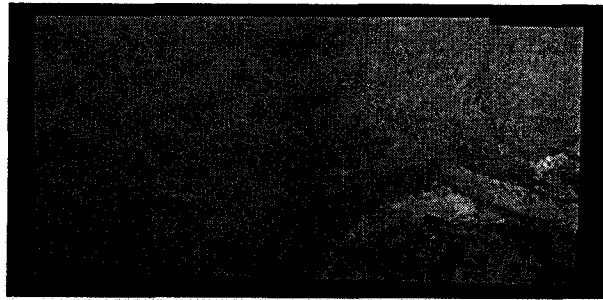
1m




(d)



(e)



(f)

1m

APPENDIX B

PSEUDOCODE

B.1 Pseudocode for Median Mosaics Based Illumination Correction

This part is identified in Section 3.1 and Section 4.1.

//Input file: a.) transformation record: TransformFile, and b) underwater video: Video

//Output file: a) CorrectedVideo, and b) CorrectedMosaic

//Functions: a) medianmosaic – mosaic images using median method,

b) getframenum – read the video file and count the number of frames,

c) getframesize – extract the dimensions of the frame,

d) getframe – extract the frame from the video file according to the frame
number,

e) getprojected – extract the back-projected image from the mosaic according
to the transformation record file,

f) gettrend – calculate the trend of the image,

g) outputvideo – compose the frames into video file,

h) mosaic – mosaic the frames according to the transformation record file.

//Pseudocode:

Image MedianMosaic = medianmosaic (TransformFile, Video);

Int FrameNum = getframenum (Video);

(*Int* W, *Int* H) = getframesize (Video);

```

Detrend Detrend = detrend (detrendOrder = 2, W, H);
For (i = 0; i < FrameNum; i++)
{
    Image Frame = getframe (Video, i);
    Image Back = getprojected (MedianMosaic, i, TransformFile);
    Vector FrameTrend = gettrend (Drend, Frame);
    Vector BackTrend = gettrend (Drend, Back);
    Image CorrectedFrame = Frame - FrameTrend + BackTrend;
    outputvideo(CorrectedVideo, CorrectedFrame, i);
}
Image CorrectedMosaic = mosaic (TransformFile, CorrectedVideo);

```

B.2 Pseudocode for Perspective Warping

This part is identified in Section 3.2 and Section 4.2.

//Input file: a.) transformation record: TransformFile, b) illumination corrected underwater video: Video, and c) sequential frame number for warping: B and D

//Output file: WarpedFrameD, WarpedMosaic

//Functions: a) getframes – extract certain frames from Video files according to the frame number,

b) getframesize – extract the demensions of the frame,

c) perspectivetrans – perspectively transform the coordinates,

d) getoverlap – extract the overlapping vectors according to transformation record file and frame number,

e) perspectivewarp – warp image according to their original and target coordinates,

f) mosaic – mosaic the frames according to the transformation record file.

//Pseudocode:

(Image FrameB, Image FrameD) = getframes (Video, B, D);

(Int W, Int H) = getframesize (Video);

ParaB[4] = {(0, 0), (0, H), (W, 0), (W, H)};

ParaD[4] = {(0, 0), (0, H), (W, 0), (W, H)};

Int i = 0;

While (i < iteration)

{

WarpedParaD = perspectivetrans (ParaD);

```
Image WarpedD = perspectivewarp (FrameD, WarpedDParaD);  
(Vector OverlapB, Vector OverlapD) = getoverlap (TransformFile, Video, B, D);  
Double Cost = sqrt (OverlapB - OverlapD);  
ParaD = WarpedParaD;  
If Cost < Threshold  
    Break;  
}  
Image WarpedD = perspectivewarp (FrameD, WarpedParaD);  
Image WarpedMosaic = mosaic (FrameB, WarpedD, TransB, TransD);
```

B.3 Pseudocode for Thin-plate Spline Warping

This part is identified in Section 3.3 and Section 4.2.

//Input file: a.) transformation record: TransformFile, b) illumination corrected underwater video: Video, and c) sequential frame number for warping: B and D

//Output file: WarpedFrameD, WarpedFrameB, WarpedMosaic

//Functions: a) getframes – extract certain frames from Video files according to the frame number,

b) gettrans – extract the transformation records for single frames from the file,

c) featuredetector – extract the feature coordinates from the images,

d) projecttoworld – transform the feature coordinates from the local image coordinate to the mosaic coordinate,

e) inverstrans – transform the feature coordinates from the mosaic coordinate to the local image coordinate,

f) thinplatesplinewarp – warp the images according to the original coordinates and the target coordinates,

g) mosaic – mosaic the frames according to the transformation record file.

//Pseudocode:

(Image FrameB, Image FrameD) = getframes (TransformFile, Video, B, D);

(Trans TransB, Trans TransD) = gettrans (TransformFile, B, D);

Features LocalFeatureB = featuredetector (FrameB);

Features LocalFeatureD = featuredetector (FrameD);

Features WorldFeatureB = projecttoworld (LocalFeatureB, TransB);

B.4 Pseudocode for Graph-cut in Gradient Domain

This part is identified in Section 3.4 and Section 4.3.

//Input file: a.) transformation record: TransformFile, b) illumination corrected underwater video: Video, and c) sequential frame number for warping: B and D

//Output file: Mosaic

//Functions: a) getframes – extract certain frames from Video files according to the frame number,

b) gettrans – extract the transformation records for single frames from the file,

c) getoverlaprectangle – extract the minimum overlapping rectangles from the two frames according to the transformation record file,

d) gradient – transform the overlapping rectangle patches into gradient domain,

e) graphcut – perform graph-cut on the vector,

f) inversegradient – inverse transform the gradient vector into special domain,

g) mosaic – mosaic the frames according to the transformation record file

//Pseudocode:

$(Image\ FrameB, Image\ FrameD) = getframes (TransformFile, Video, B, D);$

$(Trans\ TransB, Trans\ TransD) = gettrans (TransformFile, B, D);$

$(Image\ OverlapB, Image\ OverlapD) = getoverlaprectangle (TransformFile, Video, B, D);$

$(\text{Vector GradientB}, \text{Vector GradientD}) = \text{gradient}(\text{OverlapB}, \text{OverlapD});$

$\text{Vector Dif} = \text{GradientB} - \text{GradientD};$

$\text{Vector Mask} = \text{graphcut}(\text{Dif});$

$\text{Vector Gradient} = \text{Mask} * \text{GradientB} + \text{Mask} * \text{GradientD};$

$\text{Image Overlap} = \text{inversegradient}(\text{Gradient});$

$\text{Image Mosaic} = \text{mosaic}(\text{FrameB}, \text{FrameD}, \text{Overlap}, \text{TransB}, \text{TransD});$

B.5 Pseudocode for Graph-cut in Wavelet Domain

This part is identified in Section 3.5 and Section 4.3.

//Input file: a.) transformation record: TransformFile, b) illumination corrected underwater video: Video, and c) sequential frame number for warping: B and D

//Output file: Mosaic

//Functions: a) getframes – extract certain frames from Video files according to the frame number,

b) gettrans – extract the transformation records for single frames from the file,

c) getoverlaprectangle – extract the minimum overlapping rectangles from the two frames according to the transformation record file,

d) wavelettransform – transform the overlapping rectangle patches into the wavelet domain,

e) graphcutforsubband – separately perform graph-cut on the subbands of waveletvectors,

f) inversewavelettransform – inverse transform the wavelet vector into special domain,

g) mosaic – mosaic the frames according to the transformation record file.

//Pseudocode:

$(Image\ FrameB, Image\ FrameD) = getframes (TransformFile, Video, B, D);$

$(Trans\ TransB, Trans\ TransD) = gettrans (TransformFile, B, D);$

$(Image\ OverlapB, Image\ OverlapD) = getoverlaprectangle (TransformFile,$

Video, B, D);

(Vector WavletB, Vector WaveletD) = wavelettransform (OverlapB, OverlapD);

Vector Dif = WavletB – WaveletD;

Vector Mask = graphcut forsubbands (Dif);

*Vector Wavelet = Mask * WaveletB + Mask * WaveletD;*

Image Overlap = inversewavelettransform (Wavelet);

Image Mosaic = mosaic (FrameB, FrameD, Overlap, TransB, TransD);

APPENDIX C

PUBLICATION

This paper coauthored by Yuri, is accomplished during my research for this Masters' thesis, and it appears in OCEANS'06 MTS/IEEE Boston. The permission letter is enclosed after the paper.

Optimal Image Blending for Underwater Mosaics

Fan Gu, Yuri Rzhanov

Center for Coastal and Ocean Mapping, University of New Hampshire

24 Colovos Rd.

Durham, NH, 03824, USA

Email: {fan, yuri}@ccom.unh.edu

Abstract. Typical problems for creation of consistent underwater mosaic are misalignment and inhomogeneous illumination of the image frames, which causes visible seams and consequently complicates post-processing of the mosaics such as object recognition and shape extraction. Two recently developed image blending methods were explored in the literature: "gradient domain stitching" and "graph-cut" method, and they allow for improvement of illumination inconsistency and "ghosting" effects, respectively. However, due to the specifics of underwater imagery, these two methods cannot be used within a straightforward manner. In this paper, a new improved blending algorithm is proposed based on these two methods. By comparing with the previous methods from a perceptual point of view and as a potential input for pattern recognition algorithms, our results show an improvement in decreasing the mosaic degradation due to feature doubling and rapid illumination change.

I. INTRODUCTION

In the recent years, mosaics created from individual images acquired underwater are attracting more and more attention from marine geologists and biologists. Applications can be clearly divided into two categories: those targeting extraction of quantitative information (distances, sizes, shapes, etc.), and those attempting to create a consistent continuous image, possibly at the expense of minor local distortions. (A special category, aiming at accurate recovery of three-dimensional information about the seafloor, is capable of achieving both goals, but requires principally different approach, and has substantially higher level of complexity.)

In reality, due to limited visibility underwater, artificial and, as a consequence, spatially inhomogeneous illumination, and the parallax issues, most underwater images are fuzzy and difficult to process. In this paper, we are not concerned with the ability to measure distances and sizes as accurately as possible. Algorithms for object recognition and shape extraction are typically tolerant to scaling and insignificant distortions, but can be easily confused by feature doubling and rapid changes in illumination. Our goal is to diminish the effects of inhomogeneous illumination, which are almost always present in the case of artificial lighting, and to combine individual image frames into a single mosaic in some optimal way. Note that "optimal" may have different meanings depending on intended consumer: scientist, trying to deduce large-scale interrelationships; computer program, extracting

shapes according to some specific rule; or a high-school student learning about a deep-sea environment.

Current blending techniques can be divided in two main categories, assuming that the images have already been aligned. One approach is an optimal seam algorithm [1-3] that searches for a curve in the overlap region on which the differences between two overlapping images are minimal, and then each image is copied to the corresponding side of the seam. One simple and commonly used method is the minimum cut method which employs dynamic programming [1], but it works well only when two images are involved. As opposed to this "memoryless" approach, the graph-cut method [4, 5] was proposed that can be applied when more than two images are needed to be mosaiced. However, the seam may still be visible where brightness of neighboring original images differs dramatically.

Another category is aiming at smoothing the transition between two images. Most common blending techniques employ simple averaging of images in the overlapping regions. This results in ghosting artifacts, blurring, and visible seams that degrade the mosaic. Some improvement of this method were proposed, such as feathering or alpha blending [6] which employs the special weighting functions, multi-resolution blending [7-9] which takes advantage of the characteristics of different sub-bands, and gradient domain stitching [10-12,16], which is designed to remove sharp changes of brightness across the frame boundaries. However, blurring and ghosting effects could not be avoided due to misalignment of the underwater imagery.

In our paper, the methods mentioned above are explored in application to underwater images. Due to the complexity of underwater imagery, the defects of these methods are more apparent, and thus other practice should be considered in order to get higher quality mosaics for either post-processing or simple viewing. Our proposed blending method is using advantages of the graph-cut technique and gradient domain stitching method, and has achieved a significant improvement over the existing algorithms.

II. THEORETICAL BACKGROUND

In this section, methods of gradient domain stitching and graph-cut are highlighted and application details are introduced.

A. Gradient Domain Stitching

Computation in the gradient domain was recently used in compression of dynamic range [12], image editing [11], image inpainting [13] and separation of images to layers [14]. In [10], two approaches were proposed for image stitching in the gradient domain, and the previous spatial methods (such as feathering, pyramid blending and optimal seam) performed in gradient domain of the images were compared with their original methods. Results show an improvement in overcoming the photometric inconsistencies and small geometric misalignment between the stitched images. Performance here is similar to image editing [11], which suggests editing images by manipulating their gradients. One of the editing applications concerned is the object insertion, where an object is selected and cut from an image, and inserted into a new background image. The insertion process is done by solving the Poisson equation in the gradient field of the inserted patch, with boundary conditions defined by the background image.

Mathematically, the gradient of a two-variable function (here the image intensity function I) is at each image point a 2D vector with the components given by the derivatives in the horizontal and vertical directions, that is:

$$\nabla I = (\partial I / \partial x, \partial I / \partial y). \quad (1)$$

With some additional assumptions, the derivative of the continuous intensity function can be calculated as a function of the sampled intensity function, i.e. the digital image. For example, the gradient for digital images approximated by the forward difference:

$$\nabla I(x, y) \approx (I(x+1, y) - I(x, y), I(x, y+1) - I(x, y)) \quad (2)$$

In order to reconstruct the pixel values, integration should be performed, however, the conservativeness can rarely be achieved in this case [11]. Other methods were proposed to solve this problem such as Fourier basis function algorithm [15] which orthogonally projects the gradient values onto a finite set of ortho-normal basis functions spanning the set of integrable vector fields; another method is to search the function over the space of all 2D potential functions whose gradient is closest in the sense of least-squares. As proved in [11], the second method is equivalent to solving of the following Poisson equation:

$$\nabla^2 I = \text{div} G, \quad (3)$$

where Laplacian values of I is expressed as:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (4)$$

and the divergence of gradient vector G is:

$$\text{div} G = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} \quad (5)$$

Approximating them with the standard finite differences yields a linear system of equations, where the Laplacian of I is expressed as:

$$\nabla^2 I = I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4I(x, y) \quad (6)$$

and the divergence of G is:

$$\text{div} G = G_x(x, y) - G_x(x-1, y) + G_y(x, y+1) - G_y(x, y-1) \quad (7)$$

In solving the Poisson equation, boundary conditions were reported to be chosen differently according to the applications.

B. Graph-cut method

The graph-cut method [5] is designed to find a boundary between two images in such a way that the seam is the least noticeable. This search is formulated in terms of finding the minimum of a certain energy function. The graph-cut algorithm is based on the principles of combinatorial optimization, and has attracted a lot of attention recently due to its ability to solve problems of this type extremely effective.

Principally similar, dynamic programming method was first proposed in [1], which also incorporate seams finding process. However, specifics of implementation impose restrictions on the ways the seam is allowed to follow. This may lead to missing of good seams that cannot be modeled within the imposed structure. In addition, dynamic programming is "memoryless" and cannot explicitly improve existing seams. This gives limitations when appending new images to the existing images. Graph-cut technique overcomes these disadvantages by treating each pixel uniformly and is also able to place patches over the existing images.

Specifically, let x and y be two adjacent pixel positions in the overlap region. Let $A(x)$ and $B(y)$ be the pixel values in the same color channel coming from original and new images, respectively. The matching quality cost E can be defined between the two adjacent pixels x and y that are copied from patches A and B to be:

$$E(x, y, A, B) = \|A(x) - B(x)\| + \|A(y) - B(y)\|, \quad (8)$$

where $\|\cdot\|$ denotes an appropriate norm.

III. PROPOSED METHOD

The method proposed in this paper is overcoming the defects of the single graph-cut technique, which would have apparent seam when two images have inhomogeneous illumination, and the single gradient domain stitching, which can still cause blurring in a misaligned case. As mentioned in the previous sections, our method is based on the graph-cut in the gradient domain. Different from the method of optimal seam in gradient domain [10], which is the dynamic programming based method, graph-cut here is performed on the overall image values, and is more flexible in defining the "cut" area.

The procedure is as follows, assuming that two images I_D , I_B have already been aligned and we take only one color channel for illustration:

- 1) Following the formula of (4), we calculate the gradient values of two images, I_D , I_B , obtaining G_D , G_B .
- 2) According to the overlapping area (which in general is an irregular polygon), a boundary box is obtained, which is composed of three parts: overlapping area, and parts that have contributions from only one of the images.
- 3) Within the boundary box, execute the graph-cut technique and get the graph-cut mask, using weighting function to smooth the boundary cut and obtaining the final mask.
- 4) Fill in the boundary box with gradient values according to the mask matrix, and use it as a source term of the Poisson equation. Boundary values of the boundary box are from the original pixel values of two images given the boundary of the mask.
- 5) Reconstruct the spatial values of the boundary box by solving the Poisson equation with Dirichlet boundary conditions.
- 6) Put the corresponding reconstructed values back in the final mosaic.

In practice, the images are part of a sequence, for example, captured from a video tape. Transformations relating consecutive images are either deduced from the navigation data, or estimated from the imagery. Frames are added sequentially to already existing mosaics.

IV. EXPERIMENTS

In this section, the proposed method is applied and results are shown, then experiments using different methods and results are compared from the perceptual point of view.

In our case, size of the mosaic can rapidly grow so that a typical desktop computer cannot handle it. Addition of a new individual image frame to an existing mosaic is described in terms of interaction between two images: one, represented by a rectangular footprint, and another, bounded by a complex polygon, which in general may consist of several disjointed parts, be concave, and have holes. For simplicity of comparison, our experiments are performed on two overlapping gray-level frames the video footage courtesy of Dr. R. Vrijenhoek, MBARI.

A. Results of the proposed method

Below are the results following the performance procedures introduced in the previous section: first, the original images overlapping I_D and I_B are given in Fig. 1, then the position and dimension of boundary box is illustrated on the image of mosaics in Fig. 2. Graph-cut matrix with the narrow weighting function (Here, we give the result of three pixel wide band) is shown in Fig. 3. The mask, which is filled by the source gradient values and boundary conditions are in Fig. 4. In this case, the light gray stands for the gradient values from G_D , and the darker gray stands for the values from G_B , while the black

values in the bottom are the gradient values from G_B , but not out of the overlapping area. Efficient solution of the Poisson equation can be achieved by a variety of methods. We have chosen the direct solver from the INTEL Math Kernel Library, v.8.01.

B. Results from other methods

For comparison, we performed the following blending methods on the same images, and their results were given in Fig. 5-10:

- 1) the direct averaging blending method,
- 2) feathering method,
- 3) feathering in gradient domain,
- 4) direct graph-cut method,
- 5) graph-cut method in gradient domain without weighting function methods.

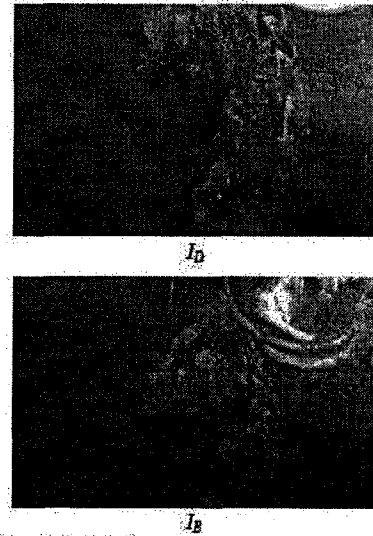


Fig. 1. Original Images I_D and I_B .

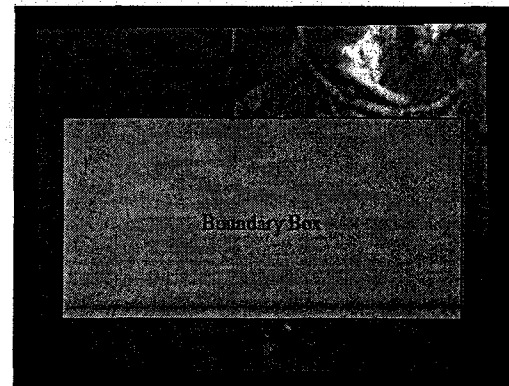


Fig. 2. The position of the boundary box.



Fig. 3. Graph-cut mask of the boundary box.

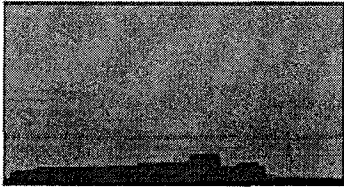


Fig. 4. Mask for gradient values filling.



Fig. 7. Feathering blending method.



Fig. 5. Graph-cut in gradient domain with a weighting function.

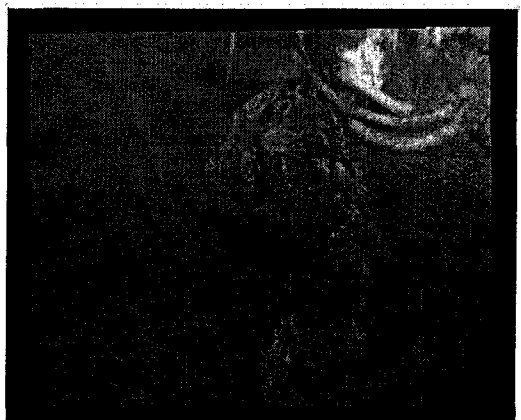


Fig. 8. Direct graph-cut method.

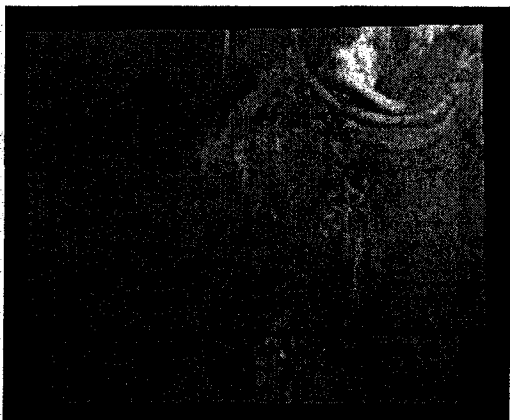


Fig. 6. Averaging blending method.



Fig. 9. Feathering in gradient domain method.

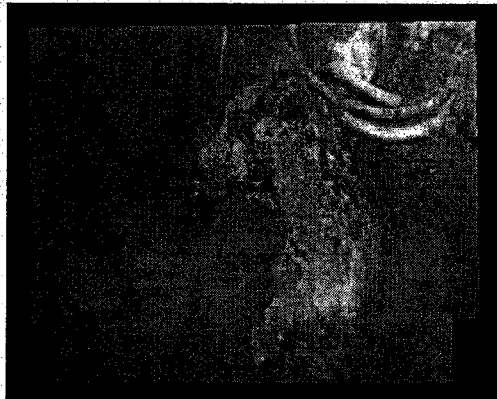


Fig. 10. Graph-cut in gradient domain without weighting function.

From Fig. 6, it can be observed that direct averaging method give rise to apparent blurring and doubling, in addition, it does not improve the illumination difference in two images. In Fig. 7, the weighted averaging seems to be improved in sense of both of these two disadvantages above, still, the seam due to the illumination difference is apparent on the right of the mosaic. In terms of decreasing the blurring and ghosting effects, graph-cut technique in Fig. 8 gives a good result, however, the seam is more apparent because, in this case, the difference of illumination is large and the seam cannot be hidden among the complex texture of the images. Fig. 9, which performs in the gradient domain with weighting function make the whole mosaic more homogeneous in illumination, however, comparing with the original images, details are blurred and not as distinctive as in the direct graph-cut method. Fig. 10 which employs graph-cut mask in the gradient domain, the mosaic illumination is more homogeneous comparing with direct graph-cut method, while the details are clearer compared to the gradient domain feathering. But there are some blocks of white along the region of cut, which are apparent artifacts. It might be due to the inconsistency of the source term along the seam. In our experiment, when using the weighting function along the cut, the effect is less apparent, as shown in Fig. 5.

V. CONCLUSIONS

Due to the artificial lighting and 3D content of imaged terrain, imagery taken underwater almost always suffers from inhomogeneous illumination and feature misalignment, when mosaiced. This causes degradation of the final product and makes it more difficult to post-process. Often used mean value averaging blending technique can hardly satisfy the demand of post processing such as feature extraction or human view leisure.

These days, a lot of blending techniques were explored in the area of image processing. Most of them fail when it comes to the underwater images, which have different specifics. We reviewed the existing popular methods and combined them in a

way to facilitate in post-processing of underwater mosaics. Specifically, we have combined the graph-cut method designed to improve on image blurring and ghosting, caused by local misalignments, and the gradient domain stitching technique, which helps with lighting inhomogeneities and exposure artifacts. Employing the graph-cut in the gradient domain eliminates their defects with the weighting band. Experimental results show the effectiveness of the proposed methods, comparing with other existing methods such as special averaging, feathering, graph-cut, and feathering, graph-cut in gradient domain. The reason for artifacts, occasionally occurring in the reconstructed process, requires further investigation.

ACKNOWLEDGMENT

The authors are grateful to Dr. R. Vrijenhoek from MBARI for the data.

REFERENCES

- [1] A. Efros and W. Freeman, "Image Quilting for texture synthesis and transfer", *SIGGRAPH2001, Proceedings of Computer Vision*, pp. 1033-1038, 2001.
- [2] D. Milgram, "Computer methods for creating photomosaics", *IEEE Trans. Computer 21*, pp. 1113-1119, 1975.
- [3] J. Davis, "Mosaics of scenes with moving objects", *CVPR*, pp. 354-360, 1998.
- [4] Y. Boykov, O. Veksler and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts", *ICCV*, 1999.
- [5] V. Kwatra, A. Schödl, I. Essa, G. Turk and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts", *ACM Transaction On Graphics*, pp. 277-286, 2003.
- [6] M. Uyttendaele, A. Eden and R. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 509-516, 2001.
- [7] P. Burt and E. Adelson, "A Multiresolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics*, V2, pp. 217-236, 1983.
- [8] S. Peleg, "Elimination of Seams from photomosaics", *Computer Graphics and Image Processing*, 16(1): pp. 90-94, 1981.
- [9] Z. Hu and S. T. Acton, "Morphological Pyramid Image Registration", *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, 2000.
- [10] A. Levin, A. Zomet, S. Peleg and Y. Weiss, "Seamless image stitching in the gradient domain", *ECCV'04*.
- [11] P. Perez, M. Gangnet and A. Blake, "Poisson Image Editing", *Proceeding of SIGGRAPH 2003*, pp. 313-318.
- [12] R. Fattal, D. Lischinski and M. Weinman, "Gradient domain high dynamic range compression", *Proceeding of SIGGRAPH 2001*, pp. 249-256.
- [13] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting", *Proceedings of ACM SIGGRAPH 2000*, pp. 417-424.
- [14] Y. Weiss, "Deriving intrinsic images from image sequences", *Int. conf. on Computer Vision*, pp. 68-75, 2001.
- [15] R. Fankot and R. Chellappa, "A method for enforcing integrability in shape from shading algorithms", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 10(4): pp. 439-451, 1988.
- [16] A. Agarwala, M. Domchaya, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin and M. Cohen, "Interactive Digital Photomontage", *ACM SIGGRAPH 2004*.

Comments/Response to Case ID: 005725B2

ReplyTo: Pub-Permissions@ieee.org

From: Anthony Ven Graitis

Date: 05/04/2007

Subject: Re: from Fan Gu
<fangu@ccom.unh.edu>
about the paper

Send To: Fan Gu

cc:

Dear Fan.

This is in response to your letter below, in which you have requested permission to reprint, in your upcoming thesis/dissertation, the described IEEE copyrighted material. We are happy to grant this permission.

Our only requirement is that the following copyright/credit notice appears prominently on the first page of each reprinted paper, with the appropriate details filled in:

© [Year] IEEE. Reprinted, with permission, from (complete publication information).

Best regards
Tony VenGraitis, IPR Specialist

IEEE Intellectual Property Rights Office
445 Hoes Lane, Piscataway, NJ 08855
Telephone: +1 732-562-3966
Fax: +1 732-562 1746
w.hagen@ieee.org
<http://www.ieee.org/copyright>

Dear Sir or Madam,

I am one of the authors of paper:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=4098825&arnumber=4099156&count=359&index=330

Optimal Image Blending for Underwater Mosaics
Fan Gu Rzhanov, Y.

Center for Coastal and Ocean Mapping, University of New Hampshire, 24
Colovos Rd, Durham, NH, 03824, USA. Email: fan@ccom.unh.edu;

This paper appears in: *OCEANS 2006*
</xpl/RecentCon.jsp?punumber=4098824>
Publication Date: Sept. 2006
On page(s): 1-5
Location: Boston, MA, USA,
ISBN: 1-4244-0115-1
Digital Object Identifier: 10.1109/OCEANS.2006.307037
Posted online: 2007-02-12 09:13:45.0

I hope to include my paper in the appendix of my masters' thesis, and
would you please forward an email giving me this authority? Thank you
very much!

Sincerely,
Fan Gu