Underwater Image Stitching

by

Saeed Hojjati

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

For years, panoramic image stitching has been an interesting problem for researchers. Several advances have been made in stitching images that are acquired outside of water, but the problem has been poorly explored for underwater images. Image stitching for underwater images can be used in numerous scientific applications in the fields of marine geology, archaeology, and biology that involve tasks such as the prospection of ancient shipwrecks, the detection of temporal changes, and environmental damage assessment. It can also be used to create virtual reality tours of zones of special interest such as underwater nature reserves.

Underwater images suffer from poor visibility conditions because of medium scattering, light distortion, and inhomogeneous illumination. This causes many image stitching techniques to fail when they are applied to underwater images. In this work, we develop a novel method for stitching underwater images. We adopt dehazing to not only improve the aesthetic quality of the images but also to enable feature detectors to accurately detect and match feature points. We also adopt guided image filtering to improve the speed of the dehazing algorithm. A novel idea proposed in this method is to use colour transfer to transform images into the same colour space in order to reduce lighting inhomogeneities and exposure artifacts. We further process our stitched results by applying a graph-cut strategy that operates in the image gradient domain over the overlapping regions to improve blurring and ghosting effects caused by local misalignments. Finally, we apply a transition smoothing method to produce more plausible results and to reduce the noticeability of the joining regions to an even higher degree.

This thesis is dedicated to
my father, my mother, and my brother.
For their endless love, support, and encouragement.

Acknowledgements

Firstly, I would like to thank my supervisor, Professor Herbert Yang, for sharing his considerable wisdom in this field, for providing invaluable guidance, and for being incredibly patient in the supervision of this thesis.

I am most grateful to my friends at the University of Alberta: Mehdi, Sina, Mohsen, Amir, Mostafa, and Saber. They made life away from home easier for me.

Finally, thanks to Mum, Dad, and Sina, for encouraging me in the pursuit of my dreams, and for providing a loving family in which to do so.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

"The *panorama* is an unbroken view of the whole region surrounding an observer" [13]. It is an alternative way of visually presenting information of a scene. Panoramic images assist users in viewing and navigating real world landscapes. Image stitching can be used to create panoramic mosaics which are often viewed interactively for applications such as virtual tourism, or to provide backdrops in films and video games for images acquired outside of water. As for underwater images, image stitching can be used to create undersea panoramic views which can be used to create virtual reality tours of zones of special interest, like shipwrecks or underwater nature reserves. In addition to that, it can also come in handy for researchers in marine geology and biology to study underwater habitat.

Underwater images suffer from poor visibility because of the medium scattering and light distortion. Most traditional computer vision methods cannot be applied directly in underwater images due to the challenging environmental conditions and the different light attenuation [55].

Attenuation caused by light that is reflected from a surface and is deflected and scattered by particles as well as substantial reduction of the light energy because of absorption by the medium make analyzing underwater images diffi-

cult. The random attenuation of the light scattered back from the water along the viewing direction considerably degrades the scene contrast. This causes underwater images to appear *hazy*.

Moreover, there is rarely any significant light beyond 200 meters (656 feet) under the ocean [32]. The ocean is divided into three zones based on depth and light level. The upper 200 meters (656 feet) of the ocean is called the euphotic, or "sunlight" zone. This zone contains the vast majority of commercial fisheries and is home to many protected marine mammals and sea turtles. Only a small amount of light penetrates beyond this depth. Below 200 meters, artificial illumination required to capture undersea images causes the illumination to be spatially inhomogeneous. Hence, these images are difficult to process.

Due to the aforementioned reasons stitching underwater images can be a more challenging task compared to the that of land images. Indeed, many existing image stitching techniques fail when they are applied to underwater images. On the other hand, the problem of underwater image stitching is poorly explored by researchers and is the motivation of this research.

1.2 Contributions

The main contributions of this thesis are listed below:

- Develop a novel method for stitching underwater images which suffer from poor visibility conditions as well as inhomogeneous illumination and feature misalignment.

- Adopt dehazing to not only improve the aesthetic quality of the final result but also to enable feature detectors to accurately detect and match feature points for underwater images, which is what they are normally incapable of.

- Apply colour normalization to reduce lighting inhomogeneities and exposure artifacts.
- Apply a graph-cut method to improve image blurring and ghosting caused by local misalignments.
- Adopt guided image filtering to improve the speed of the dehazing algorithm.

1.3 Outline of Thesis

The remainder of the thesis is organized as follows. Chapter 2 provides background and reviews the literature on image stitching and dehazing. In chapter 3 we describe the design of an image stitcher that is capable of stitching underwater images. Chapter 4 includes experiments and the evaluation of our method. In chapter 5 we present conclusions and ideas for future work.

Chapter 2

Background and Related Work

2.1 Image Stitching

Image stitching or *image mosaicing* is the process of combining multiple photographic images with overlapping fields of view to produce a panorama or high-resolution image. Exact overlaps between images and identical exposures in order to produce seamless results are required by most common approaches to image stitching [59].

2.1.1 Components of Image Stitching

The stitching algorithm consists of two main components: image registration, and blending. Image pairs are compared to find the translations that can be used for the alignments of images during image registration. After image registration, images are blended to form a single image [1]. These main components are briefly discussed below.

Registration

Image registration is the process of aligning the images which are captured from different viewpoints and is the core of any mosaicing procedure. It serves the purpose of creating geometric correspondence between images [1].

Blending

Blending is applied across the stitch to make the seams less apparent. There are many different methods for blending among which *alpha feathering* is the most commonly used. It takes the weighted average of two images [12]. The weighting function is usually a ramp. At the stitching line the weight is half and half while away from the stitching line one image is given more weight than the other. It works well in the case where image pixels are well-aligned and the only difference between two images is overall intensity shift. However, if the images are not well aligned, the disagreements will show in the blended image [1]. Another popular method is the Gaussian pyramid. This method merges the images at different frequencies and filters them accordingly. The lower the frequency band, the more it blurs the boundary. The Gaussian pyramid blurs the boundary while preserving the pixels away from the boundary. It does not work well, however, if the two images are at significantly different intensity levels [1].

2.1.2 Image Stitching Approaches

The main approaches for image stitching are the direct and feature-based techniques [1]. The direct techniques directly minimize pixel to pixel dissimilarities, while feature-based techniques work by extracting a sparse set of features and then matching them to each other.

Direct Techniques

The direct techniques work by comparing all the pixel intensities of the images with each other. They minimize the sum of absolute differences between overlapping pixels or use other available cost function. Due to the comparison of each pixel window to others, these methods are computationally expensive. Furthermore, they are not invariant to image scale and rotation [1]. There

are many techniques proposed for image stitching based on direct methods such as the Fourier Analysis Technique [5] or the unifying framework for fine optimization of cost functions proposed by Baker and Matthew [3].

Direct methods have the advantage of making optimal use of the information available in image alignment.

Feature-Based Techniques

The comparison of all features in one image against all features in the other using one of the local descriptors may seem to be the simplest way to find all corresponding feature points in an image pair. However, being quadratic in the expected number of features makes it impractical for many applications [1]. For image stitching based on feature-based techniques, the main steps required are image feature extraction, registration, and blending.

The first step is to establish correspondences between points, lines, edges, corners, or other geometric entities. A robust detector must be invariant to image noise, scale, translation, and rotation transformations. Many feature detectors have been proposed over the years such as Harris [23], SIFT [41], SURF [4], FAST [53], PCA-SIFT [30], and ORB [31].

The well-known SIFT (Scale Invariant Feature Transform) is very robust but the high computation time makes it unsuitable for some real-time applications. The Harris corner detector uses a normalized cross-correlation of intensity values to match them. However, it is not invariant to scale changes and cross-correlation. SURF (Speeded Up Robust Features) [4] improves the computation time of SIFT using an integral image to compute local gradient faster on an image. Recently, binary feature descriptors have received a lot of attention. ORB [31] is one of the binary feature descriptors. It is a combination of the FAST (Features from Accelerated Segment Test) keypoint detection and the BRIEF (Binary Robust Independent Elementary Feature)

keypoint descriptors algorithm which is modified to handle oriented keypoints. ORB is scale and rotation invariant and robust to noise and affine transformations. It is extremely fast while sacrificing little on performance accuracy.

Feature-based methods have the advantage of being more robust against scene movement that has occurred in the image. They are potentially faster and are capable of recognizing panoramas by automatically discovering the adjacency relationships among an unordered set of images [1]. These features are best suited for fully automated stitching of panoramas. Feature-based methods rely on accurate detection of image features. Correspondences between features lead to computation of the camera motion which can be tested for alignment. In the absence of distinctive features, this kind of approach is likely to fail.

Early feature-based methods seemed to get confused in regions that were either too textured or insufficiently textured. The features would often be distributed unevenly over the images thereby failing to align image pairs [59]. Furthermore, establishing correspondences relied on simple cross-correlation between patches surrounding the feature points which did not work well when the images were rotated or had foreshortening due to homographies [59]. Today however, feature detection and matching schemes are remarkably robust and can even be used for known object recognition from widely separated views. Features not only respond to regions of high "cornerness", but also to "blob-like" regions, as well as to uniform areas. Furthermore, because they operate in scale-space and use a dominant orientation (or orientation invariant descriptors), they can find matches in images that differ in scale, orientation, and even foreshortening [59].

2.1.3 Image Stitching Model Based on Feature-Based Techniques

A complete image stitching model based on feature-based techniques is discussed in this section. This model consists of four stages: image acquisition, feature detection and matching, RANSAC estimation, and image blending. In the following subsections each stage is described in detail.

Image Acquisition

Image acquisition is the first stage of any computer vision system. It can be defined as the action of retrieving an image from some sources. After the image has been taken, various different methods of processing are applied on the image so as to perform different vision tasks which are required in today's image making. If the image is not acquired satisfactorily then the subsequent tasks may not perform well, even if some image enhancement technique is applied to the images. Acquired images are assumed to have enough overlapping that the stitching can be done.

Feature Detection

Feature detection is the second step and the main stage in image stitching process. Features are the elements in the input images to be matched. Instead of looking at the image as a whole, some special points in the image are selected and a local analysis is performed on those [1].

Feature detection is an important part of many computer vision algorithms. The speed at which features are detected is important in many image processing applications, such as visual SLAM (Simultaneous localization an mapping), image registration [7], 3D reconstruction, and video stabilization in which corresponding image features are matched between multiple views [1]. The detected feature points or corners need to be unambiguous so that the

correspondence between views can be computed reliably. Real-time applications need the process of feature detection, description, and matching to be fast as well [1].

Corners are one type of feature to match in an image pair. They can be matched to give quantitative measurement. The features of corners are more stable over changes of viewpoint [1]. Besides, the intensity of a corner is significantly different from pixels in its neighbourhood.

On the other hand, there are local feature descriptors that describe a position in an image in terms of its local content. They are robust to small deformations and localization errors, and help us find the corresponding pixel locations in images which capture the same amount of information about the spatial intensity patterns under different conditions [1].

A local feature detector should not only be invariant to translation, rotation, scale, affine transformation, and the presence of noise and of image blur, but also robust to occlusion, clutter, and illumination changes [1]. There should also be enough points to represent the image in a time efficient setting.

There are many feature descriptors such as SIFT [41], SURF [4], HOG [11], GLOH [45], PCA-SIFT [30], Pyramidal HOG (PHOG), and Pyramidal Histogram of visual Words (PHOW).

Feature Matching

After extracting features and their descriptors from the input images, the next step is to establish some preliminary feature matches between the images. The algorithm consists of two parts. The first part is the selection of a matching strategy which determines which correspondences are passed on to the next stage to be further processed. The second part consists of choosing efficient data structures and algorithms. A full traversal search may seem to be the more consistent search strategy but the computation time is too large. One of

the more efficient options is a K-D tree based algorithm called the Best-Bin-First (BFF). It identifies the nearest match with high probability in an efficient time setting. It uses a modified search ordering for K-D tree algorithm so that bins in the feature space are searched in the order of their closest distance from the query location [39]. The candidate match for each keypoint is found by identifying its nearest neighbour, which is defined as the keypoint with the minimum Euclidean distance from the given descriptor vector. This method can quickly and efficiently find corresponding matches for each feature point.

Homography Using RANSAC

After computing an initial set of feature correspondences, we need to find a set that produces a high-accuracy alignment. One possible approach is simply computing a least squares estimate, or by using a robustified version of least squares. However, it is often better to find a good starting set of inlier correspondences, i.e., points that are all consistent with a particular motion estimate. The most common approaches for this task is RANdom SAmple Consensus (RANSAC) [17]. It starts by selecting a random subset of k correspondences, which is then used to compute a motion estimate. Then it counts the number of inliers that are within a specific distance of their predicted location. The random selection process is repeated for a pre-defined number of times and the sample set with the largest number of inliers is kept as the final solution.

Compositing

The last step is to decide how to represent the final stitched image. When the number of stitched images is not too high, the common approach is to select one of the images as the reference and then warp all of the other images into the reference coordinate system. The result is called a *flat* panorama in which

straight lines still remain straight (which is often a desirable attribute) [1] . However, there are other projecting layouts which might be used for specific application of which *cylindrical* panorama is another well-known method. In order to build a cylindrical panorama, we require a sequence of images taken by a camera placed on a leveled tripod. If the camera focal length is given, each perspective image can be warped into cylindrical coordinates. Forward warping and inverse warping are two types of cylindrical warping. In the former, the source image is mapped onto a cylindrical surface, but holes may exist in the destination image because some pixels may never get mapped there. In the latter, each pixel in the destination image is mapped to the source image [1].

2.1.4 Scale-Invariant Feature Transform

Scale-Invariant Feature Transform was proposed by Lowe [41]. The algorithm is used to detect and describe the local features of the image. The features detected by this algorithm are not only accurate and stable but also invariant toward scale and rotation. It is widely used for a multiple of applications such as object recognition [41], robotic mapping and navigation [56], 3D modeling [62], image stitching [8], gesture recognition [22], match moving [66], video tracking [66], and individual identification of wildlife [65].

Scale-Space Extrema Detection

In this section the scale-space theory is used to determine the keypoints that are, in other words, the interesting points. In order to detect the keypoints we first consider an image, say $I(x, y)$, and convolve that image with a Gaussian filter, $F(x, y, k\sigma)$, at varying scales. The resulting images at different scales are subtracted in order to get the Difference of Gaussian (DoG) as shown in figure 2.1.

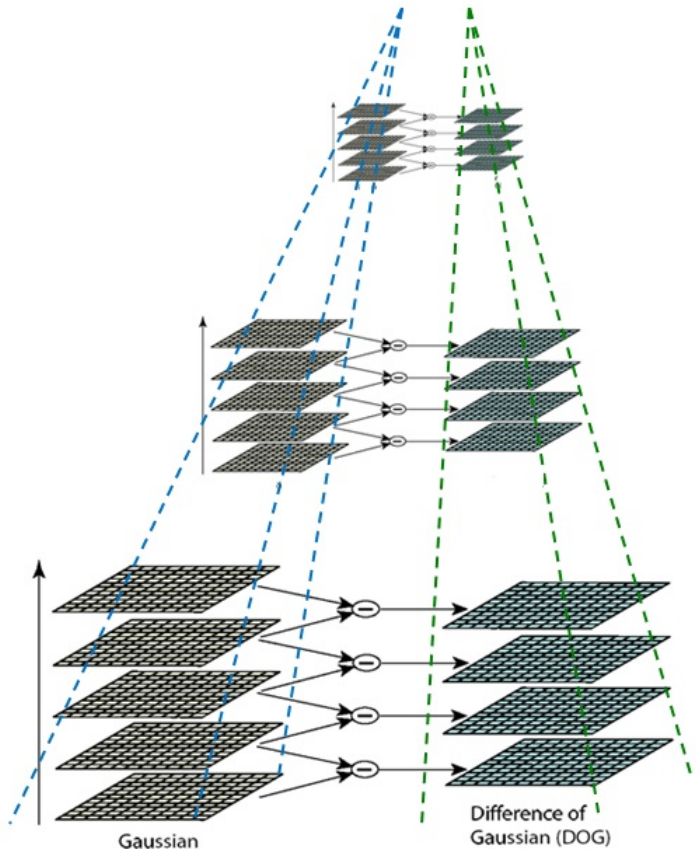


Figure 2.1: DoG at varying octaves

DoG helps to remove the problems that arise due to keypoint localization. It acts as an effective tunable band pass filter and extracts a range of components which can be used as keypoints. DoG can be used as an approximation for the Laplacian operator. The convolution of Gaussian filter is represented in equation 2.1 and DoG is represented using equation 2.2.

$$L(x, y, k\sigma) = F(x, y, k\sigma) \otimes I(x, y) \tag{2.1}$$

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, u, k_j\sigma) \tag{2.2}$$

$L(x, y, k\sigma)$ is the convolution of the original image I(x,y) with the Gaussian blur $G(x, y, k\sigma)$ at scale $k\sigma$.

Keypoint Localization

Keypoints are nothing but the local maxima or minima of the DoG images across scales and are selected after taking the DoG. Each pixel from a DoG image is compared with its 8 neighbouring pixels from the same scale, whereas the remaining 9 pixels are at the plane lying above and 9 below it each at a different scale. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint. The 26 neighbouring pixels for the candidate keypoint are shown in figure 2.2.

This keypoint detection step is a variation of one of the blob detection methods developed by Lindeberg. It works by detecting scale-space extrema of the scale normalized Laplacian [36, 37], i.e., detecting points that are local extrema with respect to both space and scale, in the discrete case by comparisons with the nearest 26 neighbours in a discretized scale-space volume [61]. The difference of Gaussians operator can be seen as an approximation to the Laplacian, with the implicit normalization in the pyramid also constituting a discrete approximation of the scale-normalized Laplacian [38, 61].

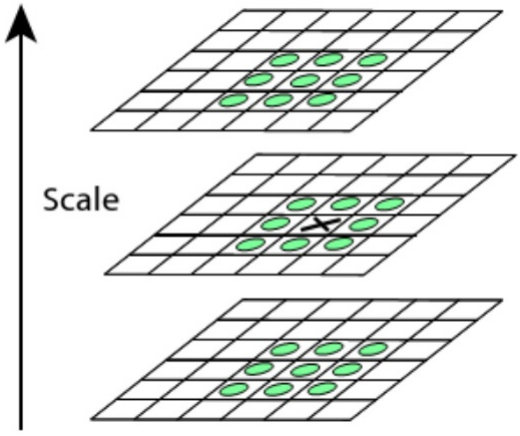


Figure 2.2: Neighbouring pixels with which key point is compared

Interpolation of nearby data is used to accurately determine the position

for each candidate keypoint. The initial approach was to just locate each keypoint at the location and scale of the candidate keypoint [40]. However, in the new approach, the interpolated location of the extremum is calculated, which improves matching and stability[41]. The quadratic Taylor expansion of the DoG scale-space function is used to determine the accurate position and scale of feature points [61] is given by

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x, \tag{2.3}$$

where D and its derivatives are evaluated at the candidate keypoint and $x = (x, y, \sigma)$ is the offset from this point. In order to determine the location of the extremum, $\hat{x}$, the derivative of this function must be taken and set to zero. If the offset $\hat{x}$ is larger than 0.5, it means that the extremum lies close to another candidate keypoint, in which case, the keypoint is changed and the interpolation is performed at that point. Otherwise, the estimate for the location of the extremum is obtained by adding the offset to its candidate keypoint [61].

Orientation Assignment

In this step, each keypoint is assigned one or more orientations depending on the local image gradient directions [61]. As the keypoint descriptors can be represented relative to this orientation, this step is essential to achieve invariance to image rotation.

First, the Gaussian-smoothed image $L(x, y, \sigma)$ having scale σ is taken to make all the computations scale-invariant [61]. For an image sample image $L(x, y)$ at scale σ, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$ are computed as follows:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{2.4}$$

$$\theta(x, y) = \arctan 2(L(x, y+1) - L(x, y-1), L(x+1, y) - L(x-1, y)) \tag{2.5}$$

The computation of magnitude and direction for the gradient are done at every pixel in the neighbouring region around the candidate keypoint in the Gaussian-blurred image L. An orientation histogram of 36 bins is created, with each bin covering 10 degrees [61]. Each sample in the neighbouring window is weighted by its gradient magnitude and a Gaussian-weighted circular window with 1.5 tine θ to the scale of the candidate keypoint, and then added to a histogram bin. The peaks of the histogram correspond to dominant orientations. After the histogram is completely filled, the orientations corresponding to the highest peaks and local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple orientations being assigned, an additional keypoint is created having the same location and scale as the original keypoint for each additional orientation [61].

Keypoint Descriptor

During the previous steps, the locations of the keypoints were found and orientations and scales were assigned to them which ensured invariance to location, scale, and rotation. The next step is to compute a descriptor vector for each keypoint that is highly distinctive and invariant to remaining variations such as illumination or 3D viewpoint [41].

First, a set of orientation histograms is created on 4×4 neighbourhoods with 8 bins each. These histograms are computed from magnitude and orientation values of samples in a 16×16 region around the keypoint such that each histogram contains samples from a 4×4 subregion of the original neighbourhood region. The magnitudes are then weighted by a Gaussian function with σ to 1.5 of the width of the descriptor window [61]. The descriptor now becomes a vector from all the values of these histograms. Since there are 4×4 histograms that each has 8 bins, the vector has 128 elements for each keypoint [41]. The diagram showing generation of feature vectors is shown in figure 2.3.

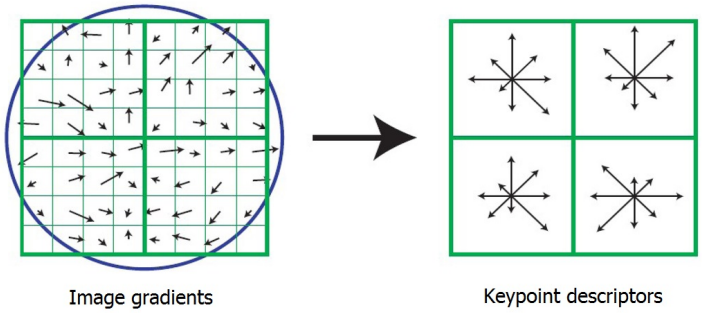


Figure 2.3: Generation of feature vectors

Finally, the feature vector is modified to reduce the effects of illumination change. The vector is first normalized to unit length. A change in image contrast in which each pixel value is multiplied by a constant will multiply gradients by the same constant, so this contrast change is canceled by vector normalization [41]. A brightness change in which a constant is added to each image pixel does not affect the gradient values as they are computed from pixel differences. Therefore, the descriptor is invariant to affine changes in illumination [41]. However, non-linear illumination changes might still occur due to camera saturation or illumination changes that affect 3D surfaces with differing orientations by different amounts [41]. This can cause a large change in relative magnitudes for some gradients but are less likely to affect the gradient orientations. Therefore, the influence of large gradient magnitudes is further reduced by thresholding the values in the unit feature vector to be no greater than 0.2, and then normalizing to unit length. This ensures a greater emphasis on the distribution of orientations rather than matching the magnitudes for large gradients [41].

2.1.5 Stitching of underwater images

One of the earliest systems used to automate the task of constructing underwater images was presented by Haywood in [25]. In their work, no feature extraction is performed and mosaicing is done by snapping images at known

positional coordinates. The task of warping images is easy in this case since the registration is known beforehand. Marks et al. develop a completely autonomous mosaicing system in [44]. Their method is column-based and uses a four-parameter semi-rigid motion model. In the paper titled "Video mosaicing along arbitrary vehicle paths" [18], mosaicing is performed by applying smoother-follower techniques to reduce image alignment error. In their method, the registration between images is computed by correlating binary images, after going through a *signum of Laplacian of Gaussian* filtering process, which reduces the effect of inhomogeneous illumination. Negahdaripour et al. [48] estimate motion from seabed images through a recursive estimation of optical flow. They expand their method to function in the presence of intensity variations and underwater medium effects in [50]. They further introduce a direct method for motion estimation in [49], which is applied to mosaicing in [63]. Gracias and Victor Santos [21] present a global alignment solution for underwater mosaicing using video-based imagery. Even though their global mosaic is constructed with a subset of images displaying significant inter-image motion, the feature matching is performed with high overlap (basically, the homography between two images with low overlap is calculated as the concatenation of video-rate homographies). Other methods in underwater mosaicing have made use of image corners and gray-level pixel-correlation to detect correspondences [20].

In order to eliminate the parallax artifacts, depth information estimation–based methods have been developed. Firoozfam et al. [16] designed an underwater panoramic imaging system for 3D scene reconstruction with conical distributive multicamera based on depth information. It can generate good-quality panoramas in underwater observations at short distance but, unfortunately, it cannot meet the real-time requirement due to the time-consuming depth estimations.

2.2 Dehazing of Underwater Images

The haze caused by light that is reflected from a surface and is deflected and scattered by water particles, the ambient light in the medium (referred to as *airlight*), the substantial reduction of the light energy due to absorption, and the colour change due to varying degrees of light attenuation for different wavelengths are all reasons that make the task of analyzing underwater images challenging [10]. The random attenuation of the light is the main cause of haze appearance while the fraction of the light scattered back from the water along the line of sight causes underwater images to lose contrast [42]. For example, in figure 2.4 the haze is caused by light scattering and the bluish tone is caused by different degrees of colour attenuation for different wavelengths.

Figure 2.4: A hazy underwater image with a bluish tone. This photo is acquired from [10] and is part of an underwater footage filmed by Bubble Vision Company.

Haze is a natural phenomena which hinders the quality of underwater images. Dehazing serves to improve the aesthetic quality of images as well as to

improve the data quality for scientific data collection and computer vision applications [9]. Various techniques have been proposed to remove the haze from underwater images and to enhance their quality. Some of them are discussed further.

In the paper titled "Recovery of Underwater Visibility and Structure by Polarization Analysis" [55], the authors analyzed the physical effects of visibility degradation. They have shown that the main degradation effects are associated with partial polarization of the light. They propose an algorithm for recovering good visibility in images of scenes based on a couple of images taken through a polarizer at different orientations. A distance map of the scene is also generated as a by-product [55]. In addition, the noise sensitivity of the recovery is also analyzed. Their proposed algorithm results in an improvement of scene contrast and colour correction and doubles the underwater visibility range. However, this method requires taking several pictures, which makes it unsuitable for single-image dehazing task.

Iqbal et al. present an underwater image enhancement method using an integrated colour model [28]. Their approach is based on slide stretching. First, contrast stretching is used to equalize the colour contrast in the images and second, saturation and intensity stretching of HSI is applied to increase the true colour and solve the problem of lighting [28]. The blue colour component in the image is controlled by the saturation and intensity in order to create the range from pale blue to deep blue. The contrast ratio is therefore controlled by decreasing or increasing its value [28].

In the paper titled "Low Complexity Underwater Image Enhancement Based on Dark Channel Prior" [64], an efficient and low complexity underwater image enhancement method based on dark channel prior is proposed. The method employs a median filter to estimate the depth map of image. Moreover, a colour correction method is applied to enhance the colour con-

trast [64]. The presented approach requires less computing time, enhances underwater images more effectively, and is suitable for implementing on the surveillance and underwater navigation for real-time purposes.

Chiang and Chen propose an algorithm called Wavelength Compensation and Dehazing [10] to enhance underwater images by a dehazing algorithm, to compensate for the attenuation discrepancy along the propagation path, and to take the influence of the possible presence of an artificial light source into consideration. In the first step, the depth map, i.e., distances between the objects and the camera, is estimated and the foreground and background within a scene are segmented. The light intensities of the foreground and background are then compared to determine the possible presence of artificial illumination. After compensating for the effect of artificial light, the haze phenomenon and discrepancy in wavelength attenuation along the underwater line of sight are corrected. Next, the water depth in the image scene is estimated according to the residual energy ratios of different colour channels in the background light. Finally, based on the amount of attenuation corresponding to each light wavelength, colour change compensation is conducted to restore colour balance. Their method significantly enhances visibility and displays superior colour fidelity in the images.

In the paper titled "Underwater Image Dehazing Using Joint Trilateral Filter" [57], a novel method for enhancing underwater images is proposed. The proposed underwater model compensates for the attenuation discrepancy along the propagation path. A fast joint trilateral filter is presented which not only removes overly dark fields of underwater images by refining depth map, but also acts as an edge preserving smoothing operator which shows better performance near the edges [57]. It also has the advantage of fast and non-approximate constant computational complexity which is independent of filtering kernel size. The enhanced images are characterized by reduced

noise level, improved quality, enhanced edges, and better exposure levels for dark regions. However, the method does not consider the possible presence of artificial illumination and its influence and in some cases, enhanced images still look dark.

Chapter 3

Proposed Method

3.1 Overview of the Proposed Method

An overview of the proposed algorithm is presented below:

Algorithm: Underwater Image Stitching

I. Dehaze input images
   (i) Estimate the atmospheric light
   (ii) Estimate the transmission map
   (iii) Refine the transmission map using guided filtering
   (iv) Recover the scene radiance

II. Normalize the colour of the input images

III. Detect keypoints

IV. Match keypoints

V. Estimate homography with matched keypoints

VI. Project onto a surface

VII. Use graph cut to find the optimal cut for stitching the images.

VIII. Use linear blending to smooth remaining seams

3.2 Single Image Haze Removal

The model often used in computer vision to describe a hazy image is as follows [60, 15, 46, 47]:

$$I(x) = J(x)t(x) + A(1 - t(x)), \tag{3.1}$$

in which I is the observed intensity, J is the scene radiance, A is the global atmospheric light, and t is the medium transmission describing the portion of the light that is not scattered and reaches the camera. The goal of haze removal is to recover J, A, and t from I [26].

The first term, $J(x)t(x)$, is called *direct attenuation* [60], and the second term is called *airlight* [60]. Direct attenuation describes the scene radiance and its decay in the medium, while airlight is caused by previously scattered light and leads to a shift of the scene colour [26]. The transmission t can be expressed as [26]:

$$t(x) = e^{-\beta d(x)}, \tag{3.2}$$

in which β is the scattering coefficient of the atmosphere. It indicates that the scene radiance is attenuated exponentially with the scene depth d [26].

3.2.1 Dark Channel Prior

The dark channel prior is based on He et al.'s [26] observations on haze-free land images: in most non-sky patches, at least one colour channel has very low intensity in some pixels. For an image J, the dark channel, J^{dark} is defined to be

$$J^{dark}(x) = \min_{c \in \{r,g,b\}} (\min_{y \in \Omega(x)} (J^c(y))), \tag{3.3}$$

where J^c is a colour channel of J and $\Omega(x)$ is a local patch centered at x [26]. He et al. demonstrated through several statistical experiments that the intensity of J^{dark} is low and tends to be zero for non-sky regions [26]. This observation is called dark channel prior. Using this prior with the hazy imaging model,

we can directly estimate the thickness of the haze and recover a high quality haze-free image. Moreover, a high quality depth map can also be obtained as a by-product of haze removal.

3.2.2 Haze Removal Using Dark Channel Prior

Estimating the transmission

We first assume that the atmospheric light, A, is given, and later on an automatic method to estimate it is presented. It is further assumed that the transmission in a local patch $\Omega(x)$ is constant [26]. Taking the min operation on equation 3.1 we have [26]:

$$\min_{y\in\Omega(x)}(I^c(y)) = \tilde{t}(x)\min_{y\in\Omega(x)}(J^c(y)) + (1-\tilde{t}(x))A^c. \tag{3.4}$$

Taking the min operation among three colour channels, the above equation can be rewritten as [26]:

$$\min_c(\min_{y\in\Omega(x)}(\frac{(I^c(y)}{A^c})) = \tilde{t}(x)\min_c(\min_{y\in\Omega(x)}(\frac{J^c(y)}{A^c})) + (1-\tilde{t}(x)). \tag{3.5}$$

According to the dark channel prior, the dark channel of the haze-free radiance, J, should be zero [26].

$$\min_c(\min_{y\in\Omega(x)}(\frac{J^c(y)}{A^c})) = 0. \tag{3.6}$$

Putting equation 3.6 into equation 3.5, the transmission $\tilde{t}$ can be estimated simply by [26]:

$$\tilde{t}(x) = 1 - \min_c(\min_{y\in\Omega(x)}(\frac{(I^c(y)}{A^c})). \tag{3.7}$$

Soft Matting

The estimated transmission map contains some block effects due to the assumption we made earlier that the transmission in each local patch is constant. In this step, the transmission map is refined. The haze imaging equation (3.1)

has a similar form with the image matting equation. Therefore, a soft matting algorithm [33] can be used to refine the transmission map [26]. Soft mattting allows us to find the globally optimal alpha matte by solving a sparse linear system of equations.

In order to obtain the refined transmission map, $t(x)$, the following cost function is minimized [26]:

$$E(t) = t^T L t + \lambda (t - \tilde{t})^T (t - \tilde{t}). \tag{3.8}$$

in which L is the Matting Laplacian matrix proposed by Levin [33], and λ is a regularization parameter. The first term is the smoothness term which encodes the colour line model and the second term is the data term which encodes the information of the transmission [26]. The constraint weight, λ , is a small value (10^{-4} in [26]).

In order to obtain the optimal t the following sparse linear system needs to be solved [26]:

$$(L + \lambda U)t = \lambda \tilde{t}, \tag{3.9}$$

in which U is an identity matrix of the same size as L.

The transmission map before and after the refinement can be seen in figure 3.1. Figure 4.3a is the soft matting result after using figure 4.3b as the data term. As we can see, the refined transmission map captures the sharp edge discontinuities and outlines the profile of the objects [26].

Recovering the scene radiance

With the refined transmission map calculated, the scene radiance can easily be recovered according to equation 3.1.

$$J(x) = \frac{I(x) - A}{t(x)} + A. \tag{3.10}$$

(a) Estimated coarse transmission map (b) Refined transmission map

Figure 3.1: The transmission map before and after refinement for a land based image (acquired from [26]).

Estimating the Atmospheric Light

At this point, the only unknown parameter in equation 3.1 is the atmospheric light A. The dark channel is used in order to estimate the atmospheric light. The top 0.1% of the brightest pixels in the dark channel are chosen, and among those, the pixel with the highest intensity in the input image I is selected as the atmospheric light [26].

3.2.3 Transmission Refinement Using Guided Filtering

As an alternative to soft matting [26], after obtaining the coarse transmission map, guided image filtering [27] can also be used to refine the transmission.

Guided Image Filtering

Guided filter is a type of edge preserving smoothing operator which filters the input image under the guidance of another image [27, 51]. By denoting the input image as p, the guidance image as I, and the filtering output as q, the local linear model of guided filter assumes that q is a linear transform of the guidance I in a window ω_k centered at pixel k, so that mathematically we have [51]:

$$q_i = a_k I_i + b_k, \forall i \in \omega_k, \tag{3.11}$$

in which (a_k, b_k) are some linear coefficients assumed to be constant in window ω_k. We use a square window of radius r. This local linear model ensures that q has an edge only if I has an edge because $\nabla q = a\nabla I$. This model has been shown useful in image matting [33], image super-resolution [67], and haze removal [26].

A guided filter seeks coefficients (a_k, b_k) that minimize the difference between the output q and the input p. For a window ω_k we minimize the following cost function [27]:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2), \tag{3.12}$$

in which ϵ is a regularization parameter. The solution to 3.12 can be given by linear regression [27, 14]:

$$a_k = \frac{\frac{1}{|\omega|}\sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \tag{3.13}$$

$$b_k = \bar{p}_k - a_k \mu_k. \tag{3.14}$$

Here, μ_k and σ_k^2 are the means and variance of I in ω_k, $|\omega|$ is the number of pixels in ω_k and $\bar{p}_k = \frac{1}{|\omega|}\sum_{i \in \omega_k} p_i$ is the means of p in ω_k [27].

The final filtering output is given by [27]:

$$q_i = \frac{1}{|\omega|} \sum_{k: i \in \omega_k} (a_k I_i + b_k) = \bar{a}_i I_i + \bar{b}_i, \tag{3.15}$$

where $\bar{a}_i = \frac{1}{|\omega|}\sum_{k \in \omega_i} a_k$ and $\bar{b}_i = \frac{1}{|\omega|}\sum_{k \in \omega_i} b_k$ [27].

With this modification ∇q is no longer a scaling of ∇i, because the linear coefficients $(\bar{a}_i, \bar{b}_i)$ vary spatially [27]. However, since $(\bar{a}_i, \bar{b}_i)$ are the outputs of an average filter, their gradients should be much smaller than that of I near strong edges [27]. In this situation we can still have $\nabla q \approx \bar{a}\nabla I$, meaning that abrupt intensity changes in I can be mostly maintained in q [27]. As can be seen locally, the output q captures similar details from the guidance I (by virtue of the local linear model), while globally, the impression of the output q

should be similar to the input p (due to the minimization of the cost function) [51].

Transmission Refinement Using Guided Filter

In [27], one of the applications of the guided filter is to refine the coarse transmission map obtained by dark channel prior [51]. The raw transmission map is refined through filtering under the guidance of the hazy image [27]. For guided filter, the guidance image I, the input p, and the filtering output q play similar roles as the input image, the trimap, and the alpha matte in the closed-form matting framework [51, 33].

The output of a guided filter proves to be one Jacobi iteration in optimizing of the cost function in [27]. The expected value of the constraint weight λ is 2 for a guided filter, which implies that the filtering output is loosely constrained by the input image p [51]. Therefore, a guided filter is applicable to transmission refinement in [26], since, in soft matting, the refined transmission map should also be loosely constrained by the coarse transmission map [51].

The guided filter produces visually similar results to what we obtained using soft matting while reducing the processing time of a sample 600×400 colour image from over 10 seconds as reported in [26] to about 0.1 second [27].

3.2.4 Colour Contrast Enhancement

The image requires balanced colour values for the RGB components to achieve a good visual quality. We use a simple yet efficient colour enhancement method proposed by Iqbal et al. [29] for this purpose. In order to equalize the RGB values, we fist calculate the average values of each colour component, namely

R_{avg}, G_{avg}, and B_{avg}.

$$\begin{aligned} R_{avg} &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I_R(i,j)}{M \times N} \\ G_{avg} &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I_G(i,j)}{M \times N} \\ B_{avg} &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I_B(i,j)}{M \times N}, \end{aligned} \tag{3.16}$$

where M and N are the dimensions of the image.

Since underwater images have high values of blue colour as compared to that of other colours, we can use the value of the blue colour to increase the value of the green and red colours in order to balance the colour values of the RGB components. In order to do that, we set the blue colour channel as a target mean and the other two colour channels are multiplied as follows to match the target [29]:

$$\begin{aligned} \tilde{I_R}(i,j) &= \frac{B_{avg}}{R_{avg}} I_R(i,j) \\ \tilde{I_G}(i,j) &= \frac{B_{avg}}{G_{avg}} I_G(i,j). \end{aligned} \tag{3.17}$$

3.3 Image Stitching

After using dehazing to improve the aesthetic quality of images as well as to improve data quality for the task of feature detection, SIFT [41] is used to find and match features between images.

The next step is to estimate the planar transformation between two different views of the same flat scene which can be described by means of a *planar homography* matrix [24, 43].

Assuming point p to be a point in a 2D plane P in 3D space, and $x_1, x_2 \in \mathbb{R}^3$ to be its projections into two different images I_1 and I_2. Also assuming the coordinate transformation between the two frames to be:

$$X_2 = RX_1 + T, \tag{3.18}$$

in which $X_1, X_2 \in \mathbb{R}^3$ are the 3D coordinates of point p relative to camera frames 1 and 2. The two projections of p in images I_1 and I_2, namely x_1 , x_2 satisfy the epipolar constraint [24]:

$$x_2^T E x_1 = x_2^T \widehat{T} R x_1 = 0, \tag{3.19}$$

where E is the essential matrix, containing information about translation T and orientation R between two camera frames, and $\widehat{T}$ is the skew-symmetric matrix codifying position T [43].

However, for points on the same plane P, their images will share an extra constraint that makes the epipolar constraint alone no longer sufficient.

Assuming $N = [n_1, n_2, n_3]^T \in \mathbb{S}^2$ to be the unit normal vector of the plane P with respect to the first camera frame, and d to be the distance between the plane P and the optical center of the first camera, we have:

$$\begin{aligned} N^T X_1 &= n_1 X + n_2 Y + n_3 Z = d \\ \Leftrightarrow \frac{1}{d} N^t X_1 &= 1, \forall X_1 \in P. \end{aligned} \tag{3.20}$$

By substituting equation 3.20 into 3.18 we get:

$$X_2 = R x_1 + T = R X_1 + T \frac{1}{d} N^T X_1 = (R + \frac{1}{d} T N^T) X_1. \tag{3.21}$$

So the *(planar) homography matrix* H is defined as follows:

$$H = R + \frac{1}{d} T N^T \in \mathbb{R}^{3\times 3}. \tag{3.22}$$

And it denotes a linear transformation from X_1 to X_2 as:

$$X_2 = H X_1. \tag{3.23}$$

However, due to the inherent scale ambiguity in the term $\frac{1}{d}T$, H can at most be recovered from the ratio of the translation T scaled by distance d.

From

$$\lambda_1 x_1 = X_1, \lambda_2 x_2 = X_2, \tag{3.24}$$

we have:

$$\lambda_2 x_2 = H\lambda_1 x_1 \Leftrightarrow x_2 \sim H x_1. \tag{3.25}$$

where $\sim$ indicates equality up to a scale factor.

The homography matrix H can be computed from the correspondences that we found in the previous steps and allows the description of 2D transformations between image pairs.

Even though the assumption that the scene is planar is rarely true in practice, the homography matrix can still be used to model the transformation between images when the magnitude of camera translation is negligible compared to the distance between the camera and the scene. The absence of this condition causes the images to display the parallax effect, i.e the difference in the apparent position of an object when viewed in two different views. We will deal with the parallax effect caused by local mis-registrations in the Post-Processing section.

3.4 Colour Correction

As discussed in chapter 1, sunlight barely penetrates beyond the *euphotic* or *sunlight* zone which extends to a depth of 200 meters (656 feet) beneath the ocean surface. This necessitates the use of synthetic illumination for capturing images beyond this depth. Synthetic lights tend to illuminate the scene in a non-uniform fashion which cause colour and intensity discontinuities and consequently, produce visible seams in our panorama. We use the colour transfer method proposed by Reinhard at al. [52] to make the colour distribution similar between the images.

This algorithm operates on images in $l\alpha\beta$ colour space proposed by Ruderman et al. [54], which is a colour space based on data-driven human perception and has the advantage of minimizing correlation between channels. After con-

verting both images to $l\alpha\beta$ colour space, the colour distribution of the source image, I_s, will be transformed to the colour distribution of the target image, I_t. According to [52], matching the mean and standard deviation along each of the three channels is sufficient to achieve the same colour distribution.

First,the mean of each of the colour channels of the source image is subtracted from the data points:

$$\begin{aligned} l_s^* &= l_s - \bar{l_s} \\ \alpha_s^* &= \alpha_s - \bar{\alpha_s} \\ \beta_s^* &= \beta_s - \bar{\beta_s}. \end{aligned} \tag{3.26}$$

Next, we scale the data points comprising the corrected image by factors determined by the respective standard deviations and add the averages computed for the source image:

$$\begin{aligned} l_{corrected}^* &= \frac{\sigma_t^l}{\sigma_s^l} l_s^* + \bar{l_s} \\ \alpha_{corrected}^* &= \frac{\sigma_t^\alpha}{\sigma_s^\alpha} \alpha_s^* + \bar{\alpha_s} \\ \beta_{corrected}^* &= \frac{\sigma_t^\beta}{\sigma_s^\beta} \beta_s^* + \bar{\beta_s}. \end{aligned} \tag{3.27}$$

Finally, we convert the result back to the RGB colour space.

3.5 Post-Processing

In order to further reduce the visible seams in the overlapped regions, the two most widely used approaches are blending and seam cutting. In the former, the entire overlapped region is blended, and in the latter, image cutting is performed between the images [2]. The most common approaches for blending include feathering [59], multi-band blending [8], and gradient domain stitching [34]. Gao et al. [19] suggested a combination of both seam cutting and blending in order to produce the best results.

3.5.1 Seam Cutting

In this step, an optimal seam between two images is found that produces the least visible seam detectable by a human observer. Moreover, there may still be localized mis-registrations present in the mosaic due to deviations from the idealized parallax-free camera model. Such deviations might include camera translation, radial distortion, the mis-location of the optical center, and moving objects [58]. Seam cutting also helps to improve image blurring and ghosting caused by mis-registrations.

In order to compute an optimal seam between two images, for each pixel in the final panorama result, its intensity should be mapped from one of the warped source images [19]. Graph-cuts optimization is used to perform this task. This segmentation is formulated as a binary labeling Markov Random Field (MRF) [35] in [19]. In order to assign each pixel p a label $l \in \{0, 1\}$, representing which warped source image its intesnsity should be mapped from, the following cost function needs to be minimized:

$$E = E_d + \lambda E_s, \tag{3.28}$$

where E_d is the data term denoting the likelihood of assigning a label to each pixel, and E_s is the smoothness term representing the cost of assigning different labels to adjacent pixels [19].

Following the formulation in [2], the data term is defined to be the gradient of a pixel at that location:

$$E_d(p, l_p) = -\nabla I_p^{l_p}, \tag{3.29}$$

where the binary label l_p decides which gradient between the two overlapped images to use [19]. This data term helps our seam cutting to cut along high-gradient edges.

The smoothness term represents discontinuities between each pair of neighbouring pixels and is defined as below [19]:

$$E_s(l_p, l_q) = (\|I^{l_p}_{(p)} - I^{l_q}_{(p)}\|^2 + \|I^{l_p}_{(q)} - I^{l_q}_{(q)}\|^2) + \beta(\|\nabla I^{l_p}_{(p)} - \nabla I^{l_q}_{(p)}\|^2 + \|\nabla I^{l_p}_{(q)} - \nabla I^{l_q}_{(q)}\|^2). \tag{3.30}$$

According to equation 3.30, if $l_p = l_q$, the smoothness cost will be zero, while if $l_p \neq l_q$, the smoothness cost will be the difference between the intensity and the gradient of the corresponding pixels in two images. Here, an intensity-based graph cut will consider that the differences between neighbouring pixels are large even in the case of an accurate registration, and therefore avoids those regions where the cut should be performed. Instead, when we use the difference between gradient vectors along the seam path, the optimal seam will be found regardless of the differences in exposure. The gradients are also less sensitive to other illumination issues such as non-uniform lighting caused by artificial illumination. Despite the benefits of the gradient term, the intensity term is kept in order to favour low photometric differences when registration is highly accurate. Therefore, we use a weighted addition between both gradient and intensity domain terms. Finally, graph-cuts optimization is used to assign the label to our MRF [6].

3.5.2 Blending

While seam cutting produces an image with no overlaps, there might still be discontinuities in intensity and colour between the images being composited. In order to reduce this, the seam is expanded by 16 pixels and an alpha blending algorithm (also called *feathering*) [58] is applied to the pixels in this expanded region. In alpha blending, each pixel in the mosaic image I is a weighted combination of the input images I_1 and I_2. We weigh the pixels in each image proportionally to their distance to the seam. This step helps to minimize seam artifacts by smoothing the transition between the images.

Chapter 4

Experiments and Results

In this chapter we talk about our evaluation method and show the results of our dehazing and stitching algorithms.

We performed our real-world experiments on a dataset provided by Pacific Biological Station, which specializes in monitoring undersea habitat. The images were acquired near Vancouver Coast, BC, at an approximate depth of 150 meters beneath the ocean surface.

We used an Intel® Core™ i7-2600 CPU @ 3.40GHz and 16GB of memory for all of our experiments.

4.1 Underwater Single Image Dehazing

First we demonstrate the performance of our dehazing algorithm. We use images that suffer from poor visibility conditions because of medium scattering and light distortion. The results of our underwater single image dehazing algorithm are shown in figure 4.1 in which soft matting and guided filtering are used to refine the transmission map.

As can be seen, the results are visually similar. The average runtime of the dehazing algorithm in our experiments for 800×600 input images is 24.3s using soft matting for transmission refinement and 0.15s using guided filtering for transmission refinement. This shows the superiority of guided filtering in

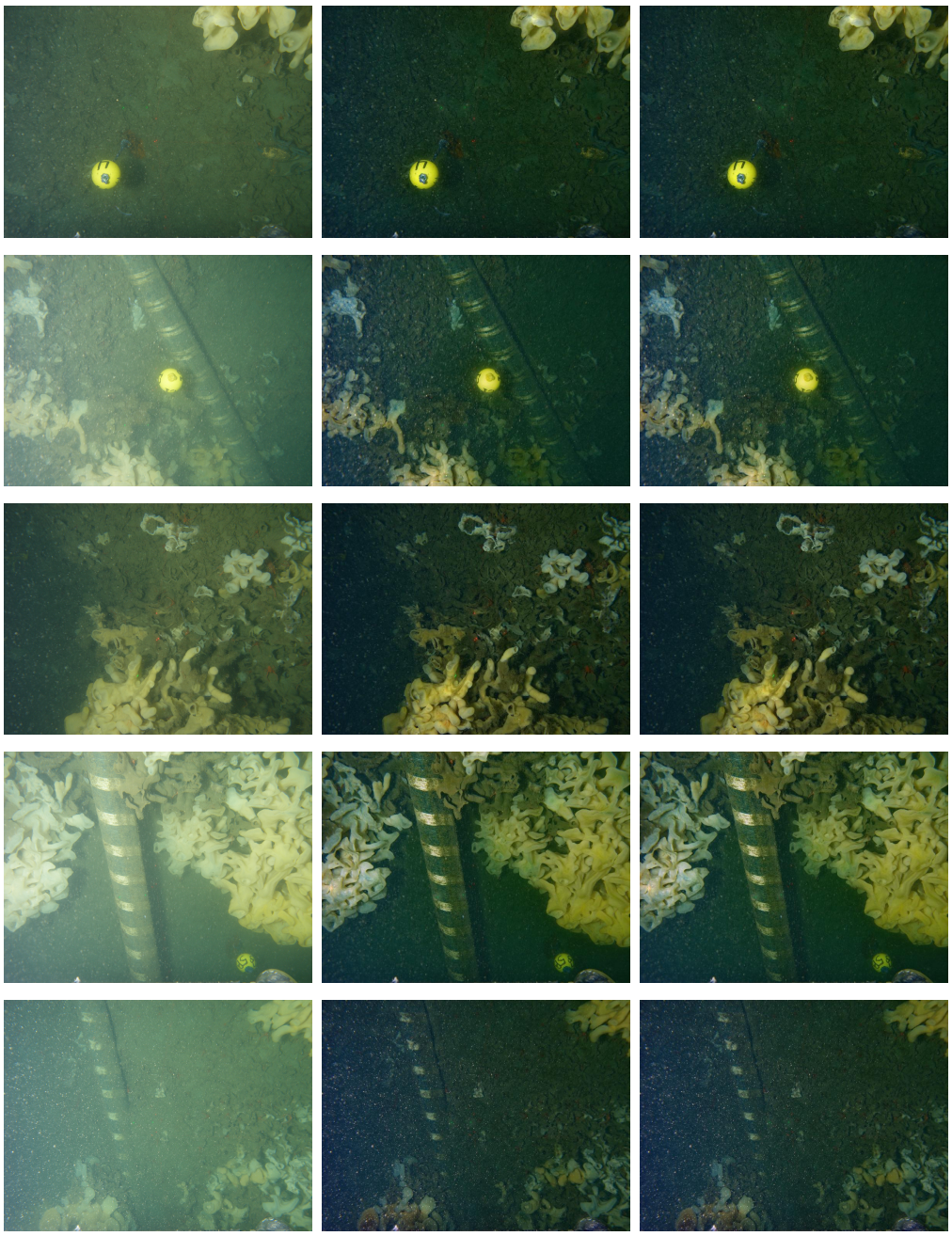


Figure 4.1: Dehazing results. The first column is the input, the second column is the obtained result using soft matting, and the third column is the obtained result using guided filtering.

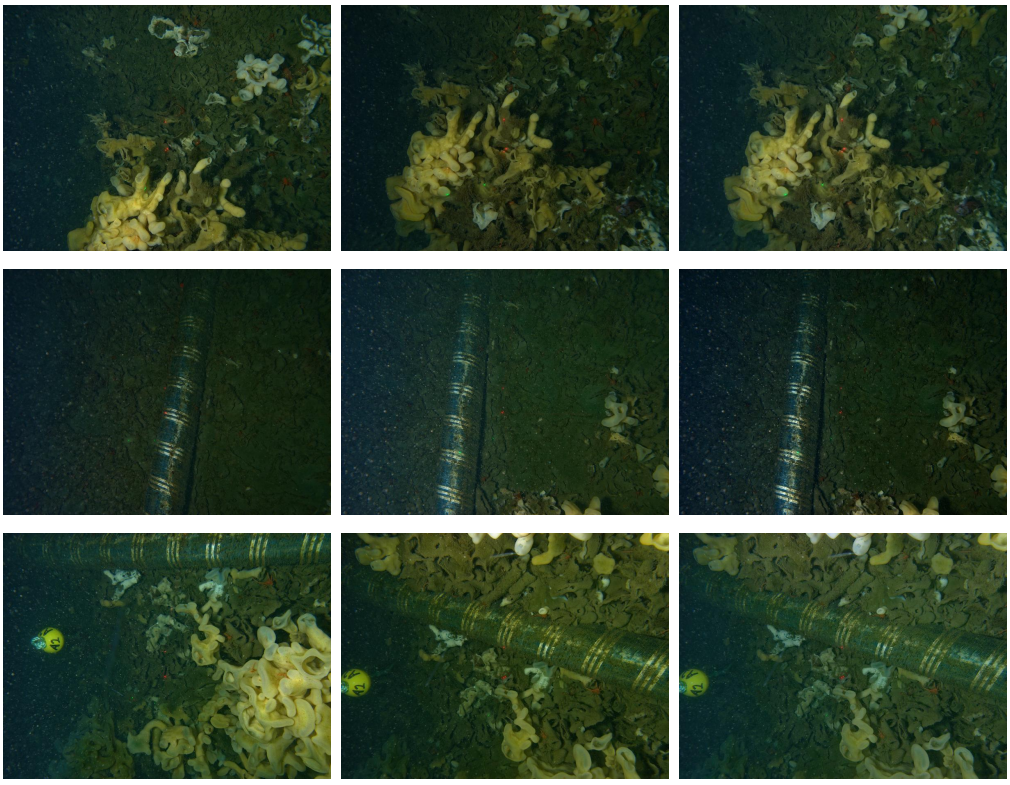
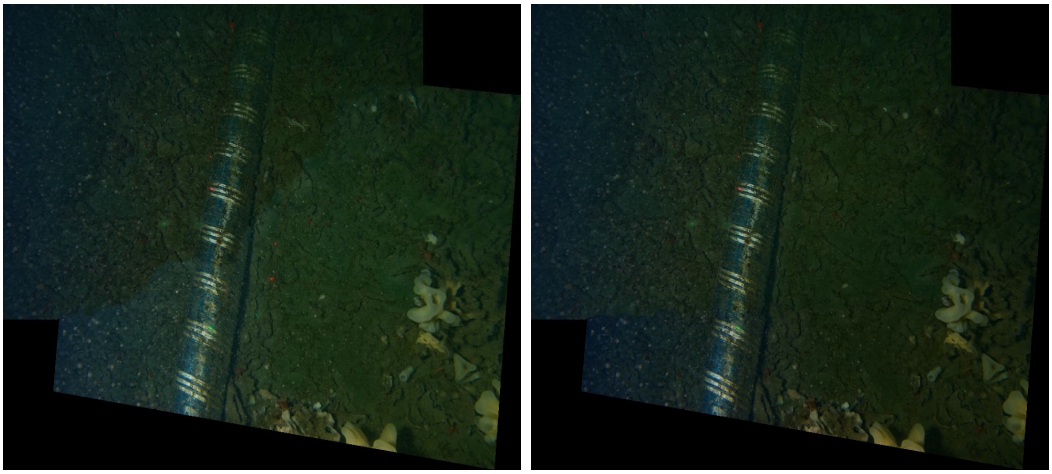


Figure 4.2: Colour transfer results. The colour of images in the second column is transferred to match the colour distributions of the images in the first column and results are shown in the third column.

terms of time complexity while producing visually similar results.

4.2 Colour Transfer Results

In this section, the results for the colour transfer algorithm are presented. In figure 4.2, the colour of the images in the second column is transferred to match the colour distributions of the images in the first column and results are included in the third column. The results of our stitching algorithm with and without the colour normalization step are included in figure 4.3 to demonstrate its importance in achieving a seamless panorama.



(a) Without colour normalization (b) With colour normalization

Figure 4.3: Stitching results with and without colour normalization

4.3 Underwater Stitching Algorithm Results

In this section, the results of our stitching algorithm are presented. In figure 4.4 the results of our experiments on real-world data are provided. As can be seen, the dehazing step enables the feature detector to detect and match features accurately. As well, blurring and ghosting effects caused by local misalignments and lighting inhomogeneities caused by artificial lighting are removed thanks to the graph-cut strategy and colour normalization, eventually leading us to seamless panorama results.

Figure 4.5 demonstrates the performance of our stitching algorithm for images taken at different illumination settings. The tank that we used is equipped with lights that have adjustable levels for different colour channels. In our first experiment, both images are captured under exactly the same lighting conditions. In our second experiment, we added the intensity of all three colour channels to capture our second input image. And finally, in the last experiment, we increased the intensity of the blue colour channel. As displayed, the algorithm shows robustness for images captured under different illumination settings.

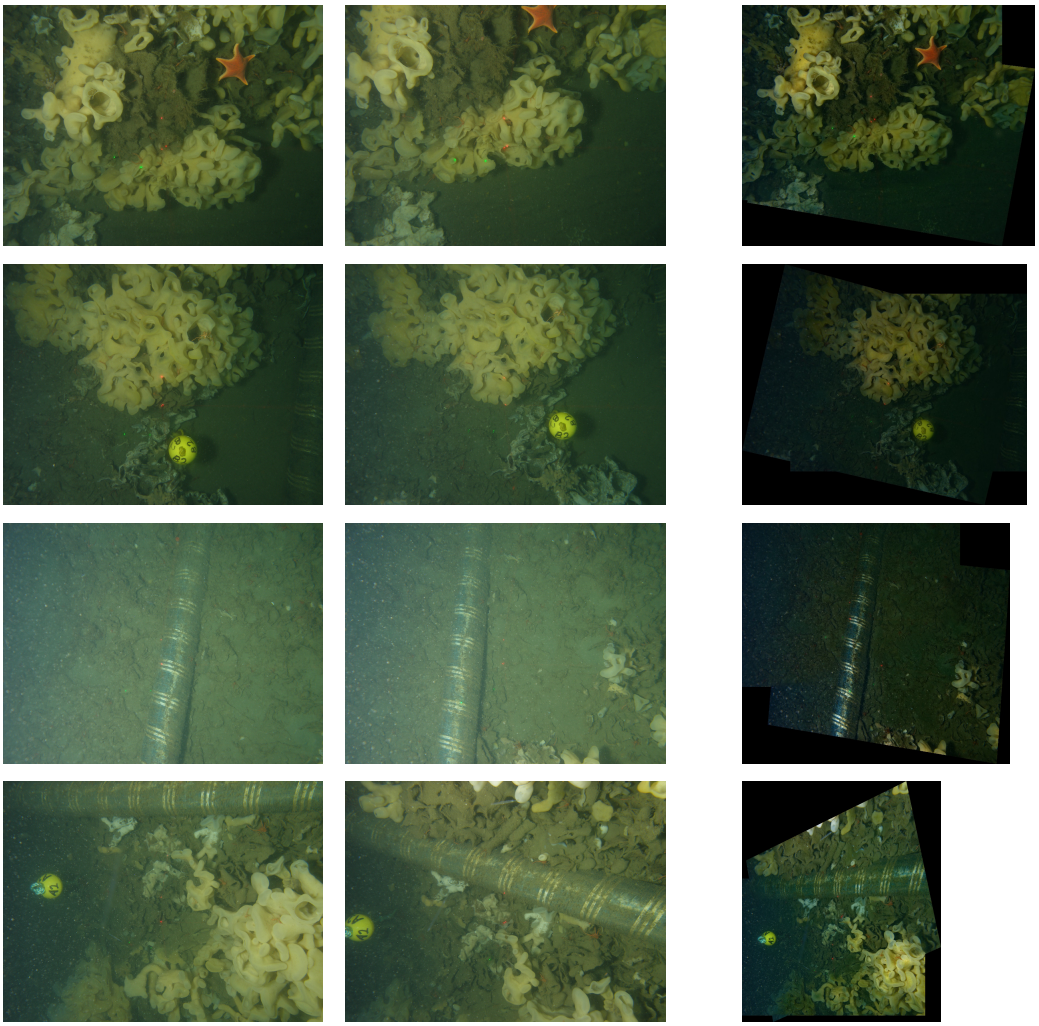


Figure 4.4: Photo stitching results for real world data. The first two columns are the inputs and the third column is the output of the stitching algorithm.

(A) Photos taken at the same illumination settings.

(B) The second input is taken under more light intensity.

(C) The second input is taken under more blue tone.

Figure 4.5: Performance of the stitching algorithm for images taken under different illumination settings

Finally, in our last set of experiments we evaluate the accuracy of our stitching algorithm. For that purpose, we use cameras with fixed positions to take pictures of the same scene after adding different amounts of milk to the tank and use the result without milk as the ground truth. We compute the difference between the homography of the stitching of hazy pictures and the ground truth and use it as the measure for the accuracy of the stitching.

$$E = \sum_{i=1}^{3} \sum_{j=1}^{3} |H_{GT(i,j)} - H_{(i,j)}|, \tag{4.1}$$

in which H_{GT} and H are the homographies of the stitching of the ground truth images and the hazy images respectively.

The results are shown in figure 4.6, and the error in the transformation for each haze level is shown in table 4.1. It can be seen that dehazing not only improves the quality of the images, but also improves the accuracy of the stitching. Furthermore, the stitching results for each of the three haze levels along with the corresponding transmission maps are included in figure 4.7.

Haze level	Level 1	Level 2	Level 3
The error in the transformation without dehazing	18.4010	25.3238	28.2983
The error in the transformation using dehazing	9.2751	13.1578	20.3883

Table 4.1: Quantitative evaluation of the stitching algorithm

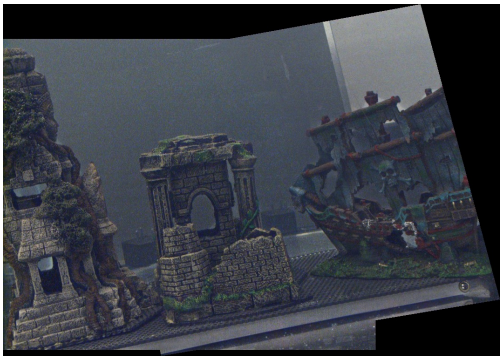
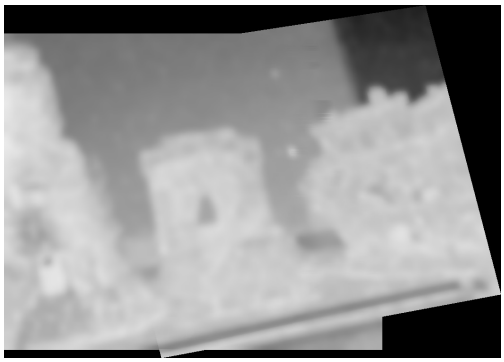
(A) Ground truth photos taken before adding milk and the stitching result (no dehazing).

(B) Photos taken after adding milk (level 1) and the stitching result using dehazing.

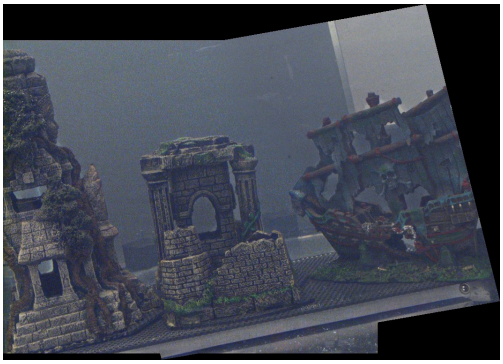
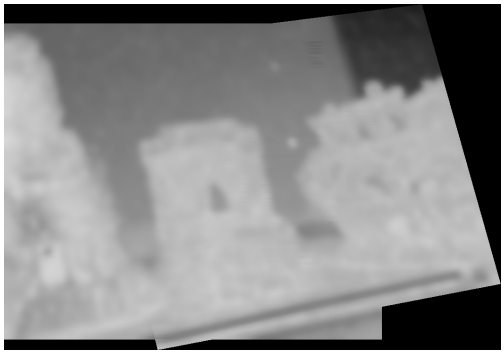
(C) Photos taken after adding milk (level 2) and the stitching result using dehazing.

(D) Photos taken after adding milk (level 3) and the stitching result using dehazing.

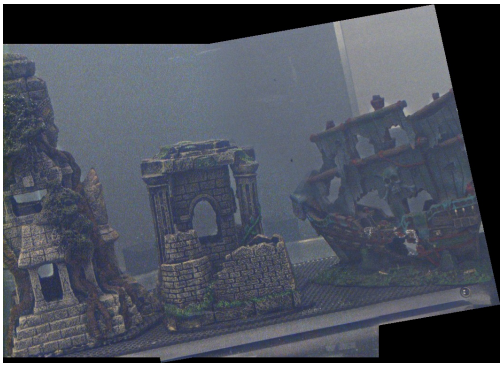
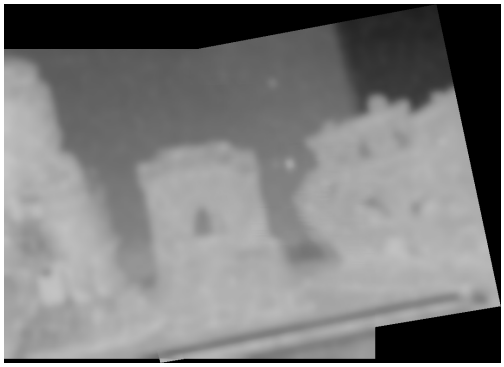
Figure 4.6: Performance of the stitching algorithm for different haze levels. The first two columns are the inputs and the third column is the output of the stitching algorithm using dehazing.

(A) The stitching result and the corresponding transmission map for haze level 1.

(B) The stitching result and the corresponding transmission map for haze level 2.

(C) The stitching result and the corresponding transmission map for haze level 3.

Figure 4.7: The stitching result and the corresponding transmission map for different haze levels.

Chapter 5

Conclusions and Future Work

This thesis has presented a novel approach for stitching images acquired underwater which is able to tackle the problems that arise when using common image stitching methods on underwater images. In the first step, dehazing is used to improve the aesthetic quality of images as well as to improve data quality for the task of feature detection. Guided image filtering is used to speed up the process of dehazing the images. Then SIFT is used to find and match features between images and a single homography per image was used to perform alignment. In the next step, a graph cuts-based seam cutting method in the image gradient domain is used to find the optimal cut between two images in order to reduce visible seams in the overlapped regions. While producing an image with no overlaps using seam cutting, we use linear blending to reduce colour discontinuities that may still exist.

A novel idea proposed in this method is to use colour normalization to transform images into the same colour space to make the stitching result even more "*seamless*".

5.1 Future Work

We conclude by identifying some avenues for future exploration:

- Compensate for *refraction*, i.e., the bending of light rays when traveling

from one medium to another.

- Apply global statistics from both images to perform dehazing.
- Apply more than one homography per image.
- Extend the method to work on multiple images to create large-scale photo mosaics and be able to perform mosaicing on unordered image datasets.
- Apply depth-dependent illumination compensation.

Bibliography

[1] Ebtsam Adel, Mohammed Elmogy, and Hazem Elbakry. Image stitching based on feature extraction techniques: A survey. *International Journal of Computer Applications*, 99(6), 2014.

[2] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions on Graphics (TOG)*, 23(3):294–302, 2004.

[3] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[5] James R Bergen, Patrick Anandan, Keith J Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Computer Vision—ECCV'92*, pages 237–252. Springer, 1992.

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[7] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[8] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[9] Nicholas Carlevaris-Bianco, Anush Mohan, and Ryan M Eustice. Initial results in underwater single image dehazing. In *OCEANS 2010*, pages 1–8. IEEE, 2010.

[10] John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength compensation and dehazing. *Image Processing, IEEE Transactions on*, 21(4):1756–1769, 2012.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[12] Yining Deng and Tong Zhang. Generating panorama photos. In *IT-Com 2003*, pages 270–279. International Society for Optics and Photonics, 2003.

[13] Oxford English Dictionary. Oxford: Oxford university press, 1989.

[14] Norman Richard Draper, Harry Smith, and Elizabeth Pownell. *Applied regression analysis*, volume 3. Wiley New York, 1966.

[15] Raanan Fattal. Single image dehazing. In *ACM Transactions on Graphics (TOG)*, volume 27, page 72. ACM, 2008.

[16] Pezhman Firoozfam, Shahriar Negahdaripour, and Caroline Barufaldi. A conical panoramic stereo imaging system for 3-d scene reconstruction. In *OCEANS 2003. Proceedings*, volume 4, pages 2303–2308. IEEE, 2003.

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[18] Stephen D Fleischer, Howard B Wang, Stelphen M Rock, and Michael J Lee. Video mosaicking along arbitrary vehicle paths. In *Autonomous Underwater Vehicle Technology, 1996. AUV'96., Proceedings of the 1996 Symposium on*, pages 293–299. IEEE, 1996.

[19] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 49–56. IEEE, 2011.

[20] Nuno Gracias and Jose Santos-Victor. Automatic mosaic creation of the ocean floor. In *OCEANS'98 Conference Proceedings*, volume 1, pages 257–262. IEEE, 1998.

[21] Nuno Gracias and José Santos-Victor. Underwater mosaicing and trajectory reconstruction using global alignment. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 4, pages 2557–2563. IEEE, 2001.

[22] Pallavi Gurjal and Kiran Kunnur. Real time hand gesture recognition using sift. *International Journal of Electronics and Electrical Engineering*, 2(3):19–33, 2012.

[23] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[24] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[25] Rodney M Haywood. Acquisition of a micro scale photographic survey using an autonomous submersible. In *OCEANS'86*, pages 1423–1426. IEEE, 1986.

[26] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2341–2353, 2011.

[27] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409, 2013.

[28] Kashif Iqbal, Rosalina Abdul Salam, Mohd Osman, Abdullah Zawawi Talib, et al. Underwater image enhancement using an integrated colour model. *IAENG International Journal of Computer Science*, 32(2):239–244, 2007.

[29] Kashif Iqbal, Michael Odetayo, Anne James, Rosalina Abdul Salam, and Abdullah Zawawi Hj Talib. Enhancing the low quality images using unsupervised colour correction method. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1703–1709. IEEE, 2010.

[30] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.

[31] AV Kulkarni, JS Jagtap, and VK Harpale. Object recognition with orb and its implementation on fpga. *International Journal of Advanced Computer Research*, 3(3):164–169, 2013.

[32] ZhongPing Lee, Alan Weidemann, John Kindle, Robert Arnone, Kendall L Carder, and Curtiss Davis. Euphotic zone depth: Its derivation and implication to ocean-color remote sensing. *Journal of Geophysical Research: Oceans (1978–2012)*, 112(C3), 2007.

[33] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, 2008.

[34] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *Computer Vision-ECCV 2004*, pages 377–389. Springer, 2004.

[35] Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

[36] T Lindeberg. Scale-space theory in computer visionkluwer academic. *Dordrecht/Norwell*, 1994.

[37] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[38] Tony Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5):10491, 2012.

[39] Haifeng Liu, Meixia Deng, and Chuangbai Xiao. An improved best bin first algorithm for fast image registration. In *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, volume 1, pages 355–358. IEEE, 2011.

[40] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[41] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[42] Huimin Lu, Yujie Li, Lifeng Zhang, Akira Yamawaki, Shiyuan Yang, and Seiichi Serikawa. Underwater optical image dehazing using guided trigonometric bilateral filtering. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2147–2150. IEEE, 2013.

[43] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[44] Richard L Marks, Stephen M Rock, and Michael J Lee. Real-time video mosaicking of the ocean floor. *Oceanic Engineering, IEEE Journal of*, 20(3):229–241, 1995.

[45] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[46] Srinivasa G Narasimhan and Shree K Nayar. Chromatic framework for vision in bad weather. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 598–605. IEEE, 2000.

[47] Srinivasa G Narasimhan and Shree K Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3):233–254, 2002.

[48] Shahriar Negahdaripour and Joel Fox. Undersea optical stationkeeping: Improved methods. *Journal of Robotic Systems*, 8(3):319–338, 1991.

[49] Shahriar Negahdaripour and Jin Lanjing. Direct recovery of motion and range from images of scenes with time-varying illumination. In *Computer Vision, 1995. Proceedings., International Symposium on*, pages 467–472. IEEE, 1995.

[50] Shahriar Negahdaripour and Chih-Ho Yu. A generalized brightness change model for computing optical flow. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 2–11. IEEE, 1993.

[51] Jiahao Pang, Oscar C Au, and Zheng Guo. Improved single image dehazing using guided filter. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011*, 2011.

[52] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, (5):34–41, 2001.

[53] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[54] Daniel L Ruderman, Thomas W Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: Implications for visual coding. *JOSA A*, 15(8):2036–2045, 1998.

[55] Yoav Y Schechner and Nir Karpel. Recovery of underwater visibility and structure by polarization analysis. *Oceanic Engineering, IEEE Journal of*, 30(3):570–587, 2005.

[56] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 2051–2058. IEEE, 2001.

[57] Seiichi Serikawa and Huimin Lu. Underwater image dehazing using joint trilateral filter. *Computers & Electrical Engineering*, 40(1):41–50, 2014.

[58] H-Y Shum and Richard Szeliski. Construction of panoramic image mosaics with global and local alignment. In *Panoramic vision*, pages 227–268. Springer, 2001.

[59] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[60] Robby T Tan. Visibility in bad weather from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[61] Wikipedia. Scale-invariant feature transform — wikipedia, the free encyclopedia, 2015. [Online; accessed 24-November-2015].

[62] Changchang Wu, Friedrich Fraundorfer, Jan-Michael Frahm, and Marc Pollefeys. 3d model search and pose estimation from single images using vip features. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[63] Xun Xu and Shahriar Negahdaripour. Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 2, pages 1412–1417. IEEE, 1997.

[64] Hung-Yu Yang, Pei-Yin Chen, Chien-Chuan Huang, Ya-Zhu Zhuang, and Yeu-Horng Shiau. Low complexity underwater image enhancement based on dark channel prior. In *Innovations in Bio-inspired Computing and Applications (IBICA), 2011 Second International Conference on*, pages 17–20. IEEE, 2011.

[65] Xiaoyuan Yu, Jiangping Wang, Roland Kays, Patrick A Jansen, Tianjiang Wang, and Thomas Huang. Automated identification of animal species in camera trap images. *EURASIP Journal on Image and Video Processing*, 2013(1):1–10, 2013.

[66] Ying Zheng, Da-Hui Li, and Ce Han. Video image tracing based on improved sift feature matching algorithm. *Journal of Multimedia*, 9(1):130–137, 2014.

[67] Assaf Zomet and Shmuel Peleg. Multi-sensor super-resolution. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 27–31. IEEE, 2002.