Restoration of Underwater Images

Seminar Report

Submitted in partial fulfillment of requirements for the degree of

Master of Technology
Computer Science and Engineering

By

Kalpesh Dusane
173050042

Guided By

Prof. Ajit Rajwade

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

May, 2018

Acknowledgements

I Would like to thank my guide **Prof. Ajit Rajwade** for counselling throughout the course of the seminar. The weekly meetings helped me understand the reference papers extensively. I would like to appreciate the insightful inputs given by **Prof. Ajit Rajwade** and **Jerin Geo James**, which help me developed the simulation for Circular ripples.

Abstract

The problem of restoring underwater images is pivotal due to its wide range of application in the study of the underwater ecosystem like coral reefs and surveillance application. More than often recorded underwater images are distorted due to various factors like impurities in water, wavy nature of water and other factors. Thus it becomes challenging task to remove these distortions. There are several methods exists which tackle some portion of the problem.

This seminar report focusses on a particular type of distortion that occurs when the scene being observed is in water and the camera is in the air (or vice versa). This distortion occurs in the form of non-rigid spatial deformations due to dynamic refraction that occurs because of the wavy nature of the water surface. We consider different models for the nature of the water surface - unidirectional cyclic waves (UCW) as well as circular ripples.

Index Terms– **Underwater image reconstruction, surface reconstruction, motion blur, circular ripples**

Contents

List of Figures

Chapter 1

Introduction

This section introduces the problem and explains why it is important to solve this problem.

1.1 Problem Definition

When a camera takes the photo of a scene which is lying beneath the water surface, most of the time the captured image is distorted. This happens because of various factors such as low contrast because of bluish colour or impurities in the water which also leads to noise and another factor can be physical characteristics of water: wave-like refraction caused by water surface, and many other factors. Thus one has to remove the distortion to better analyze the underlying image. The task is to extract the underlying original scene from the distorted image. This means one has to reconstruct the original image given the deformed image. Thus the essential problem is to *how to recover the underlying original image from the observed distorted image.*

1.2 Motivation

The need for processing underwater image is increased in recent years because of the advancement in surveillance technology. It has become very easy to capture image or video through the drones [7]. There are numerous scenarios where it is crucial to know the original information by processing underwater images. Due to a substantial rise in global warming, it has become pivotal to study the effects on the ocean and its aquatic ecosystem. Also, it is important to find the distribution of various life forms across the ocean for helping the marine biologist to decide which zone to examine. It is easy to observe the things on the land or on the surface but it becomes problematic in case of the underwater ecosystem like coral reefs [4] and other living life forms. Thus to study the effect, we have to clearly observe the changes. To do this one solution is to take the photographs by going underwater. However every time someone needs to study effects we have to send someone to take the picture which is not only tedious but also inefficient and expensive if the investigation area is far from land. Therefore it is easy to take the picture from above the water surface and then process the image to study the effect. It has application in finding the safe regions [4] for doing water related activities like boating, surfing or water skiing and other activities. There is an interesting application in the Navy [2], where submarines have to gauge the scene above the sea water level. Here, the camera is submerged in the water and the scene which has to observe is above the water surface.

1.3 Outline

This report follows certain structure where it is primarily divided into 3 major sections. In the beginning, the chapter 2 explains the experimental environment and it introduces the image formation model. Later chapter 3 examines the three different approaches to solve the problem. Thereafter, in chapter 4, we present a detailed study of the special case of underwater imaging caused by the presence of circular ripple and its challenges. Afterward, it also proposes the approaches to tackle these challenges. At last, the report demonstrates the experimental results. Conclusively, the chapter 5 proposes plans for the future work. The chapter 6 concludes this seminar report.

Chapter 2

Physical Model for Dynamic Refraction

The camera setup used is a **scaled orthographic camera**. The task is to observe the planar scene which is lying at the bottom of the water body. Here both camera and a planar scene are stationary. So image formed on the camera is because of the orthographic projections which are the projection of a 3D object out upon a plane caused by a set of parallel rays which are orthogonal to the image plane. Because of this setup, object size does not change with the change in the distance between the planar scene and the camera.

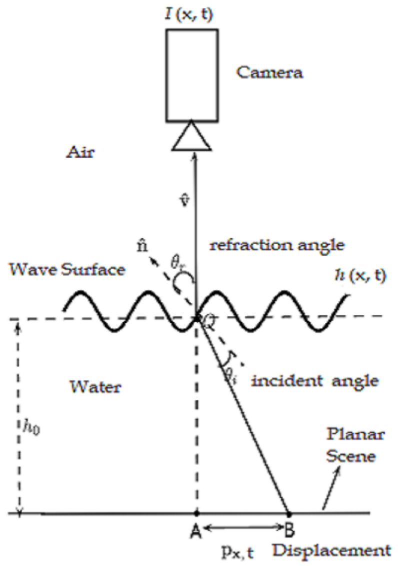

Figure 2.1: Setup of traversal of a light ray incident on the orthographic camera ”(Figure 1 [4], reproduced without permission)”

2.1 Image formation in flowing water

Consider at the bottom of the water source there is a static planar scene say $I_g(\mathrm{x})$, and from a fronto-parallel camera which was located above the water surface captures a sequence of images say $I(\mathrm{x}, t)$. As a result of the fluctuating water flow, camera records video frame $I(\mathrm{x}, t)$ which is deformed version of the static image i.e. original scene $I_g(\mathrm{x})$. These two images are related [4] by the following equation:

$$I(\mathrm{x}, t) = I_g(\mathrm{x} + w(\mathrm{x}, t)) \tag{2.1}$$

where, $w(\mathrm{x}, t) \implies$ the unknown distortion on the pixel at location x at time t.
This equation is represented by model setup shown in Fig 2.1. The problem is that we have to recover the original image $I_g(\mathrm{x})$ using a video sequence $I(\mathrm{x}, t)$ as both the state of water and the underwater scene is unknown.

Here the Fig 2.1 displays the setup where light ray is traversing from point B to the camera lens.
Some notations : $\theta_i \implies$ the angle of incidence and $\theta_r \implies$ the angle of refraction,
$h_0 \implies$ average height of the water level from the bottom,
$n_{\text{water}} \implies$ the refractive index of water, and $n_{\text{air}} \implies$ the refractive index of water,
$\mathrm{p}_{\mathrm{x},t} \implies$ Portion of displacement experienced by a light ray and it can be evaluated by considering triangle formed by points O, A and B i.e. $\triangle$ OAB and we know the angle formed by side OA and OB as $\theta_r - \theta_i$ and by taking the tangent of this angle we get as follows [5]:

$$\begin{aligned}\left\|\mathrm{p}_{\mathrm{x},t}\right\|_2 &= h_0 \tan(\theta_r - \theta_i) \\ &= h_0 \frac{(\tan\theta_r - \tan\theta_i)}{(1 + \tan\theta_r \tan\theta_i)}\end{aligned} \tag{2.2}$$

Here it is assumed that water fluctuations are small, so θ_r and θ_i will be small. Which implies that cos $\theta_r \approx 1$, cos $\theta_i \approx 1$, $1 + \tan\theta_r \tan\theta_i \approx 1$ and tan $i \approx$ sin i. Applying these approximations, above equation 2.2 can be simplified as

$$\begin{aligned}\left\|\mathrm{p}_{\mathrm{x},t}\right\|_2 &\approx h_0(\sin\theta_r - \sin\theta_i) \\ &= h_0 \sin\theta_r (1 - \frac{\sin\theta_i}{\sin\theta_r})\end{aligned} \tag{2.3}$$

From Snell's law, we get $n_{\text{water}}\sin\theta_i = n_{\text{air}}\sin\theta_r$. As the refractive index of air i.e. n_{air} is 1 and Substituting $n_{\text{water}} = n$, we get $n\sin\,\theta_i = \sin\theta_r$. Therefore, by substituting $\frac{sin\theta_i}{\sin\theta_r} = \frac{1}{n}$, equation 2.3 becomes

$$\begin{aligned}\left\|\mathrm{p}_{\mathrm{x},t}\right\|_2 &= h_0 \sin\theta_r (1 - \frac{1}{n}) \\ &= \alpha\sqrt{1 - \cos^2\theta_r}\end{aligned} \tag{2.4}$$

where, $\alpha = h_0(1 - \frac{1}{n})$ From this [4] we get,

$$\left\|\mathrm{p}_{\mathrm{x},t}\right\|_2 = \alpha \frac{\|\bigtriangledown h\|_2}{\sqrt{1 + \|\bigtriangledown h\|_2^2}} \tag{2.5}$$

where, $\bigtriangledown h \Longrightarrow$ the gradient of the water surface i.e. $\bigtriangledown h = [h_x h_y]^T$,
$h_x \Longrightarrow$ partial derivatives of $h(\mathrm{x}, t)$ with respect to x coordinate,
$h_y \Longrightarrow$ partial derivatives of $h(\mathrm{x}, t)$ with respect to y coordinates.
For small water waves, $\|\bigtriangledown h\|_2 \ll 1$ and previous equation 2.5 simplifies to

$$\left\|\mathrm{p}_{\mathrm{x},t}\right\|_2 \approx \alpha \|\bigtriangledown h\|_2 \tag{2.6}$$

It implies that displacement is proportional to the gradient of the corresponding point on the water surface [4]. So it is expressed as $\mathrm{p}_{\mathrm{x},t} \approx \alpha \bigtriangledown h(\mathrm{x}, t)$ from [5] with $\alpha = h_0(1 - 1/n) > 0$.

From Fig 2.1, we can see that because of the wavy water surface camera observes the point B instead of observing the point A. So the distortion vector expressed as

$$w(\mathrm{x}, t) = \mathrm{P}_{\mathrm{p}_{\mathrm{x},t}} \tag{2.7}$$

Where, P $\Longrightarrow$projection matrix of the camera,
$w(\mathrm{x}, t) \Longrightarrow$ "the amount of translation that a pixel x experiences on the image plane" [4].
Thus from [5] we can relate the distortion function (warping) $w(\mathrm{x}, t)$ to the height $h(\mathrm{x}, t)$ of water surface at time t :

$$w(\mathrm{x}, t) \propto \alpha \bigtriangledown h(\mathrm{x}, t) \tag{2.8}$$

From equation 2.6, it can be seen that each pixel have a particular translation which leads to image skewing effect. Thus it implies that this translation effect is **spatially varying**.

Chapter 3

Literature Survey

This literature survey explores the existing approaches which tried to solve the similar problem and analyze this approaches thoroughly so that to use them in solving the problem defined earlier.

3.1 Reconstruction of Water Surface

This method reconstructs the surface shape of water by using cues of refraction and the apparent motion of the observed pattern. The approach used in this method is similar to *shape from motion* method. However, it differs from shape from motion in context as it does not use the assumption of rigidity instead uses physical properties like optical laws and statistical motion characteristics of the object. Also, it uses refracted image to track points instead of some specific object.

Problem: Due to refraction at the water surface, the observed pattern of the object under water with waves appears to be moving. A plane whose surface normal is vertical is recognized as an average surface. For example, still water without any fluctuations(waves) will be considered as an average surface and a wave will be considered as a distortion from an average surface. This method's [3] principal objective is to reconstruct the shape of the water surface from the distorted images which were captured from above of the wavy water surface. However, while solving this problem, it retrieves the original scene which was lying at the bottom of the water body. So it can be seen as a byproduct of this method. As the original scene is unknown prior to applying this method, it can be used as a method to only recover the original underlying scene.

This method has two crucial concepts as follows:

1. There are a lot of similarities between this approach and the inverse of the ray-tracing method.
2. The underlying original scene is same as the observed image which was taken in still water, i.e. the flat water surface.

Here in the experimental setup, images were recorded in orthographic projection by a pinhole camera. There are some assumptions made in this approach such as

- To prevent any sharp separation of the image the amplitude of the wave should be small.
- Water is clean and transparent so that underlying scene will be able to completely observed. Also, it should be refractive (refractive index of water is 1.33).
- The underlying scene is planar and stationary.
- The height of the water surface from underlying planar scene, i.e. h_0 is already known.

3.1.1 Reconstruction Algorithm

The purpose of the algorithm is to reconstruct the shape of the water surface and to detect the image under the water from given input video sequence. It consists of the following steps:

1. ***Extraction of the optical flow***
   - Optical flow represents the movement of a particular point within the image from one time frame into another time frame. It is computed from the observed image sequence.
   - Any point in the optical flow diagram corresponds to vector which represents the speed and direction of the motion at that specific point.
   - To compute optical flow here we use the correlation method from various other methods.
   - Detecting the path of the optical flow of a particular point is the trajectory of that point.

2. ***Center of the Trajectory(COT)***
   - Center of the Trajectory (COT) is the average of the trajectory that is obtained in the previous step for each and every point in the observed image.
   - Consider a particular point lets say P_t in the observed deformed image at time t. From optical flow, the trajectory of P_t is obtained.
   - The COT of the point P_t can be described as the average coordinate (x, y) of the trajectory of P_t.
   - The average surface normal usually converges to zero after some time. It arises due to the mixture of sine waves model assumed in the paper [3]. However, for generalizing cases, the convergence can be achieved when it satisfies the following condition for all the points:

$$|\mathrm{COT}_{(\mathrm{using} P_1, P_2, \ldots, P_t)} - \mathrm{COT}_{(\mathrm{using} P_1, P_2, \ldots, P_{t+1})}| < threshold \tag{3.1}$$

3. ***Retrieve the underlying planar scene [optional step]***
   - The underlying scene i.e. original image can be retrieved by mapping all pixels of an observed distorted image at a certain time on their corresponding COT.
   - It is reconstructed by performing inverse transformation on each pixel of the image to its COT.

- If the only purpose to implement this approach is to recover underlying scene then do not need to implement next remaining steps.

4. ***Calculation of Gradient Vector field***
   - If the points (x_t, y_t) are known and COT (x', y') is obtained from the second step, then the direction of the vector normal to the water surface (l_t, m_t) can be computed from Image formation equations.
   - The gradient at (x_t, y_t) is calculated as

$$\begin{aligned} l_t &= \frac{(x_t - x')n}{(h_0(n-1))} \\ m_t &= \frac{(y_t - y')n}{(h_0(n-1))} \end{aligned} \tag{3.2}$$

5. ***Surface Shape Reconstruction***
   - Given the gradient vectors of each point which are computed by the previous step, the surface shape of water is reconstructed by integrating the gradient vectors over the cartesian dimensions (i.e. x and y directions).
   - This is done by the frankot-Chellappa method [1].

3.1.2 Limitations

There are some limitations to this approach such as:

- It is computationally expensive and error prone to extract the optical flow vector for all points in the image. So one solution for reducing the computation is to compute optical flows, COT's and gradient vectors for specific selected points and then calculate the gradient vector of each pixel by averaging the gradient vector of four points adjacent to the pixel. Then also it remains computationally expensive, as to compute optical flow of every pixel in the image and do processing on each pixel is computationally intensive.

3.2 UnderWater Image Restoration by Model-based Tracking

This method is similar to the previous approach, as it simultaneously estimates the shape of the water surface and recovers the underlying planar picture/pattern without using any prior information about the image or using knowledge of calibration pattern. However, the procedure to accomplish this goal is different in this approach.

3.2.1 Wave Equation

From the chapter 2, we know the relationship between distortion function and height of the water surface. "When the maximum surface fluctuation i.e. $\max_{x,t} |h(x,t) - h_0|$ is small related to h_0" [6], the water surface is regulated by the following wave equation:

$$\frac{\partial^2 h(\mathrm{x}, t)}{\partial t^2} = c^2 \bigtriangledown^2 h(\mathrm{x}, t) \tag{3.3}$$

Where, $c \Longrightarrow$ the speed of the wave which is $\sqrt{gh_0}$ as g is the gravity constant.
This equation integrates the wavy nature of water, so it can represent the generalize water surface.

3.2.2 Water Surface Synthesis using the Wave equation:

A water simulator can be build based on equation 2.8 provided the model given in section 2.1 to form the image. This simulator is generalize and can accommodate the various set of equations. Here, we use forward Euler method [6] to replicate the wave equation. As it is most basic and easy to execute and for small step size $\triangle t$ it is numerically steady.

3.2.3 A reduced model for water distortion

The number of parameters present in $h(\mathrm{x}, t)$ will multiplication of dimensions of a single video frame. Suppose video frame contains q_1 rows and q_2 columns, furthermore the number of parameters of $h(\mathrm{x}, t)$ will become $q_1 q_2$ that can lead upto quite a large number. Here, most parameter estimation methods will be unsuccessful at such a high dimensional space because of various factors like time complexity, the problem of local maxima.

Due to the smooth water surface, the structure of the $h(\mathrm{x}, t)$ can be encapsulated using a small number of dimensions. It can be achieved by using any of the model dimensionality reduction techniques like **Principal component analysis (PCA)**. As a consequence of this, it benefits in two ways for water shape estimation as follows :

1. Due to the reduction in the number of parameters, it reduces the computational time of the estimation process.

2. As a result of lower dimensional bases, it unintentionally enforces the smoothness of wave-like characteristics on $h(\mathrm{x}, t)$.

Then they divide $\mathrm{w}(\mathrm{x}, t)$ into small patches and then generate the PCA bases for these patches. Empirically, they found out that bases corresponding to the first 10 eigenvalues gives optimum result. The resulting PCA bases are $B(x) = [b_1(x), b_2(x), ..., b_9(x), b_{10}(x)]$. It is also called as **water bases**. These bases do not get affected by variation in translation.

After deriving bases, we can represent any patch of the warping $\mathrm{w}(\mathrm{x},t)$ as a linear combination of these water bases as

$$w(\mathrm{x}, t) \approx B(\mathrm{x})\ p_t \tag{3.4}$$

Where, $p_t \Longrightarrow$ coefficients corresponding to water bases.
It is insignificant to reconstructing the exact data and approximate wave-like reconstruction will also work in this method. That's why the reduction technique does not damage the performance of this method. Also, as the number of parameters increases, it may increase overhead in estimation.

3.2.4 Model-based tracking without template

For tracking water surfaces, we have to estimate the coefficients p_t for each input frame. The conventional tracking techniques are unsuitable as they required template. As in this scenario, a template cannot be used because it is not sure that any single frame is distortion-free. Also, one cannot pick any frame randomly from video frames as a template since the distortion function is not one-to-one.

The aim is to find a consistent underlying image in a manner that it can use to generate any frame in the video by warping the coefficients on to the underlying image. Consider two frames at different time period, using warping coefficients, we can obtain the undistorted image estimates. For example, in the figure 3.1 frame s and t can produce an estimate of an undistorted image by using warping coefficients p_s and p_t respectively. Suppose we derive the estimates of undistorted images as $\hat{I}(\mathrm{y}^s, s)$ and $\hat{I}(\mathrm{y}^t, t)$ from two frames which are at the time s and t respectively. A frame at time s can be denoted as $I(\mathrm{x}^s, s)$ Transformation can be expressed as

$$\mathrm{y}^s(\mathrm{p}_s) = \mathrm{x}^s + \mathrm{w}(\mathrm{x}^s, s) \approx \mathrm{x}^s + B(\mathrm{x}^s)\ \mathrm{p}_s \tag{3.5}$$

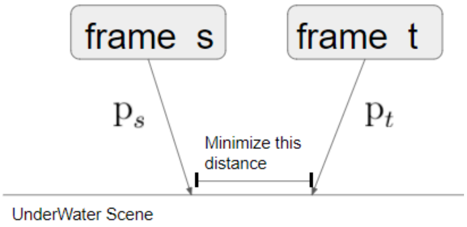

Figure 3.1: Water Tracking Approach "from Figure 8 [6], reproduced without permission"

Similarly, one can derive this equation for a frame at time t.

Thus, the objective function becomes the minimization problem, where the function has to minimize the squared difference between the estimates of undistorted frames to estimate the coefficients.

The objective function can be conceptualized as

$$J(s,t) = (Warpback(I_s, p_s) - Warpback(I_t, p_t))^2 \tag{3.6}$$

In implementation objective function is expressed in the discrete domain as

$$\hat{J}(\mathrm{p}_s, \mathrm{p}_t) = \sum_{i=1}^{n} (\hat{I}(y_i^s, s) - \hat{I}(y_i^s, t))^2 + \mathrm{symm}_t \tag{3.7}$$

The value of $\hat{I}(y_i^s, t)$ will be found by interpolating nearby samples from the particular frame t. Minimizing the equation 3.7 i.e. objective function is a non-convex optimization problem which may lead to issue of local minima. For reliable estimation, it tries to minimize the difference between coefficients instead of directly estimating two coefficients. Also, it will be robust to any translation changes. There are Two heuristics used to handle the problem of local minima. One for Long video sequences and other for Short video sequences.

Heuristics for Long video sequences:
Find the center frame of input sequence suppose center frame is c. Take initial coefficient of the frame as zero and then estimate the warping difference. Then optimize the objective function between frame c and frames c±2 using the solution of the previous step which was optimization between frame c and its adjacent neighbors. And this will repeat till optimization does not cover all the frames. This type of model help preserve the long-term connections between frames. After calculating all the warping difference, one constraint is enforced which is $\sum_i \mathrm{p}_i = 0$. This is valid because of long duration and periodic characteristics of water wave.

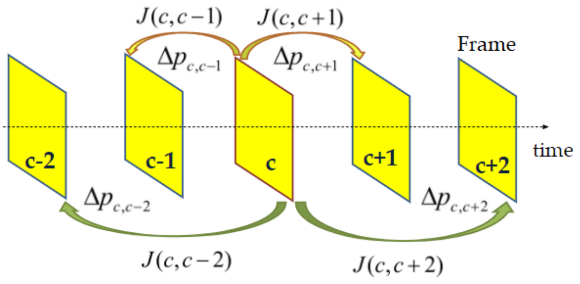

Figure 3.2: Exploiting temporal continuity in long video sequence

Heuristics for Short video sequences:
Here, the only warping difference between neighboring frames is computed. Following objective function [6] is minimized to estimate coefficients:

$$E(\mathrm{p}_1, .., \mathrm{p}_T) = \sum_{t=1}^{T-1} \left\| \mathrm{p}_{t+1} - \mathrm{p}_t - \triangle \mathrm{p}_{t+1,t} \right\|^2 + \lambda \sum_{t=1}^{T} \|\mathrm{p}_t\|^2 \tag{3.8}$$

Where, $T \Longrightarrow$ the total number of frames.
Here, the penalty term is added because for the short duration periodic characteristics of water wave may not hold true.

3.2.5 Limitations

Following are the limitations of using the model-based tracking method:

- This method required a higher frame rate to be able to reconstruct the underlying scene. Normally, frame rate required is around 125 frames-per-second (fps).
- Here, they empirically found out how many water bases required to encapsulate warping function. However, they can not find out the optimum theoretical number, so the user has to tweak it to get the desired result as the low number may not incorporate all the features of warping and the high number causes an increase in time to execute the procedure.
- If the exposure time is longer then this method does not account for motion blur which reflects in as poor quality image restoration. Experimental results in [4] shows that quality of the restored image by this method is lower than approach explain in the section 3.3 for both scenarios of cyclic waves and circular ripple.
- Does not perform well if significant refraction has happened by the water surface. It becomes the hard problem [6] because they assume that wave is smoothly fluctuating.

3.3 Image Restoration using motion blur model

There are many factors which affect the underwater scene, but we primarily focus on the effect of the dynamic refractive nature of the water surface [4]. Due to the dynamic refractive medium of water surface, the underwater scene is affected by motion blur and skew. The effect of skew is observed because of wavy nature of water. As the amount of refraction experienced at the water surface will be different for different light rays which leads to image skew. This is the **space-variant effect**.

3.3.1 Why Motion Blur?

Motion blur is predominantly caused due to relative motion between camera and scene i.e. either the scene is moving and a camera is still or camera is moving and the scene is stationary. But in our experimental setup, we assume that both camera and scene are static. Although, motion blur is experienced because of the dynamic water surface and long camera exposure. As one of the reasons for happening motion blur is when the image changes throughout the recording of single exposure due to sudden movement or long exposure. Thus, while capturing the image, the single pixel may observe light rays from multiple points from the scene during the exposure time and due to the averaging operation will cause motion blur.

3.3.2 Assumptions

Here we explore distortion caused because of **Unidirectional Cyclic Wave (UCW)**, as they are most generally occurring phenomena in water flow and no other factors are considered. For simulation purpose, we consider unidirectional cyclic wave as a sinusoidal UCWs. However, it is not a necessary condition to implement this approach and other families of waves could be considered for the UCW model. Thus water surface's mathematical equation [4] is

$$h(\mathrm{x}, t) = h_0 + A \sin\left(w_x x + w_y y + \phi(t)\right) \tag{3.9}$$

where,
$h_0 \Longrightarrow$ height of water surface from planar scene,
$A \Longrightarrow$ amplitude, $\mathrm{x} \triangleq (x, y)$ i.e. pixel location,
$w_x \Longrightarrow$ spatial frequencies in X direction,
$w_y \Longrightarrow$ spatial frequencies in Y direction,
$\phi(t) \Longrightarrow$ phase of the sine wave which depends on t as $\phi(t) \in (0, 2\pi)$. Due to this phase variation, we are able to capture dynamic nature of water.

Another assumption we make that the camera has a **long exposure** time. This is equivalent to long video duration. If exposure is short then only skewing effect will be observed; however, long exposure leads to motion blur.

3.3.3 Space Invariant distortion

As the example taken in [4] in Fig.2. It exploits the periodic nature of the UCW. It shows that every pixel experiences each variation of the gradient as one cycle is completed as shown in the figure 3.3. Also, we have previously argued in the chapter 2, the effect of skew is space-variant which implies that at a particular time instance, each pixel experiences different gradients of the water surface, which leads to different transformations at different pixels.

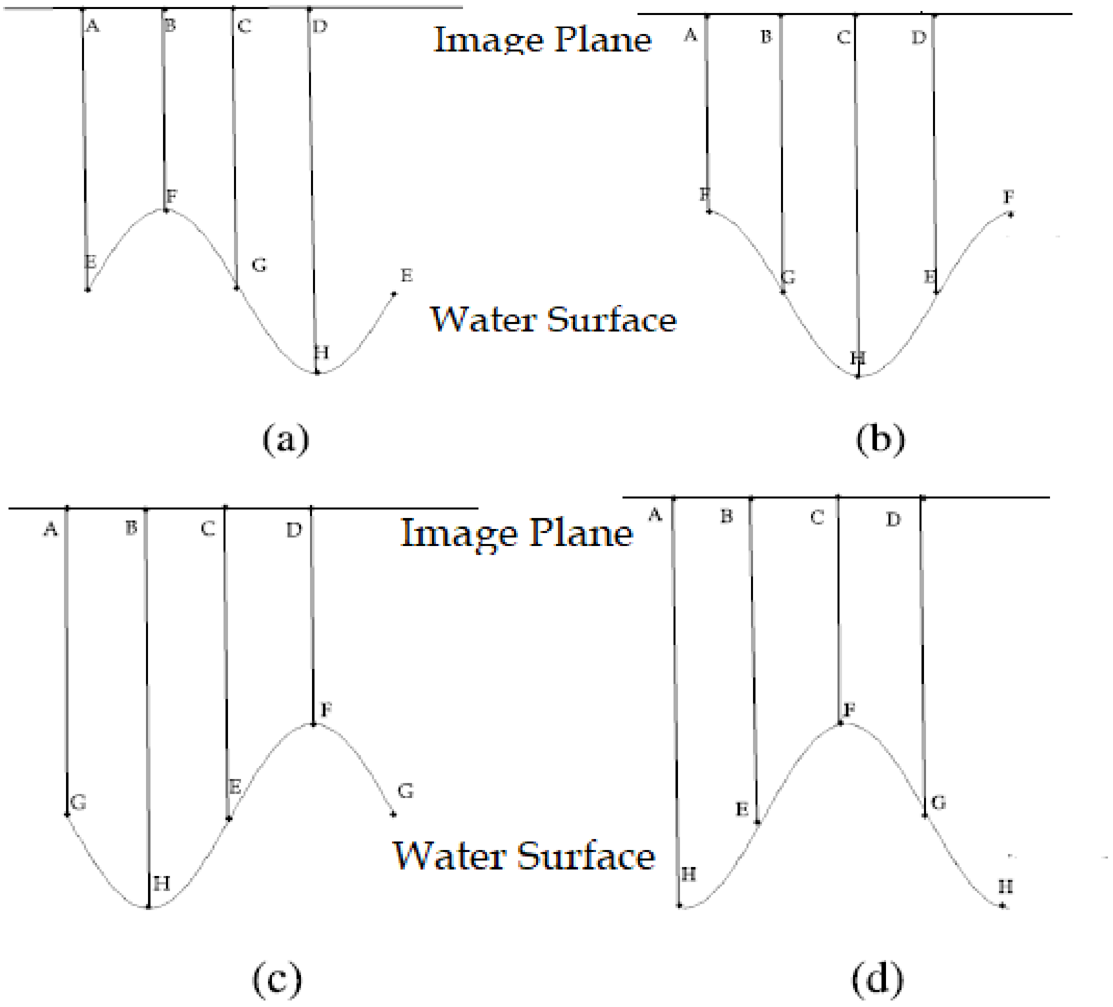

Figure 3.3: "Variation of gradients of the periodic water surface: Time (a) $T = 0$, (b) $T = t_1$, (c) $T = t_2$, and (d) $T = t_3$ (from Figure 2 [4] reproduced without permission)"

Here, the pixels A, B, C, and D observed the same set surface points E, F, G, and H in one complete cycle only in different sequence.

A motion blurred image is obtained by averaging the skewed image across different time instances during the camera exposure time. Suppose f is the image captured in still water and g is the image captured in dynamic water. As we have shown above, we can express g as the average taken during exposure time T_e of all skewed image of f. It is mathematically expressed as

$$g(\mathrm{x}) = \frac{1}{T_e} \int_0^{T_e} f(\mathrm{x} + \mathrm{w}(\mathrm{x}, t))\ dt \tag{3.10}$$

where, $\mathrm{x} \triangleq (x, y)$ i.e. pixel location. Here we are taking average so sequence in which pixel undergoes different gradient does not matter. Therefore, in UCW every pixel in image plane experienced the same set of transformation given long exposure time. This implies that blur induced is **space-invariant**. It infers that the blurred image is equivalent to a weighted average of global translations of the undistorted image f.

3.3.4 Method for recovering a distorted scene

We have to find the original undistorted image given observed blurred image. To solve this problem we use **Maximum A Posteriori Probability (MAP) estimate** formulation. Which states that

$$\textit{posteriori probability} \propto \textit{likelihood} \times \textit{prior}$$

Let I be the latent image (original image), $t_D : D \mapsto \mathbb{R}^+$ be the Point Spread Function (PSF) which map each transformation $\lambda \in D$ to a non-negative real number. This number represents the little amount of time that light ray spent in the particular position which leads to transformation λ. Motion blur is generally characterized by a PSF. As in this scenario, the single point in the image oscillates in a certain region so PSF represents the distribution of the how many times the point is at particular location in the image. Because of that, it is essential to know the PSF to recover the original image. And g be the given blurred observed image. Thus by MAP estimate, we get,

$$p(\mathrm{I}, t_D|g) \propto p(g|\mathrm{I}, t_D) \times p(\mathrm{I}) \times p(t_D) \tag{3.11}$$

where,
$p(\mathrm{I}, t_D|g) \Longrightarrow$ posterior probability,
$p(g|\mathrm{I}, t_D) \Longrightarrow$ likelihood,
$p(\mathrm{I}) \Longrightarrow$ prior on the latent image,
$p(t_D) \Longrightarrow$ prior on the PSF.

Here we are estimating the latent image and set of translation warps using above equation. Thus to compute this (posterior probability) we required three factors which can be computed as follows:

1. **Likelihood:**
   The blurred image g can be represent in terms of the PSF like

   $$g(\mathrm{x}) = \sum_{\lambda \in D} t_D(\lambda) f(H_\lambda^{-1}(\mathrm{x})) \tag{3.12}$$

   where,
   $H_\lambda \Longrightarrow$ homography matrix (which reshape the pixel x with respect to transformation λ),
   $t_D(\lambda) \Longrightarrow$ PSF of transformation λ. Also $\sum_{\lambda \in D} t_D(\lambda) = 1$.
   In general for most cases, the noise will be present. Thus the equation of the blurred image becomes

   $$g(\mathrm{x}) = \sum_{\lambda \in D} t_D(\lambda) f(H_\lambda^{-1}(\mathrm{x})) + \eta \tag{3.13}$$

   where, $\eta \Longrightarrow$ noise which is modeled as independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and standard deviation σ_0:

   $$p(\eta) = \prod_i N(\eta_i | 0, \sigma_0) \tag{3.14}$$

   where, $i \Longrightarrow$ pixel index.
   To reduce the ringing artifact we work in the gradient domain along with intensity. Therefore, we get the likelihood as

   $$p(g|\mathrm{I}, t_D) = \prod_{\delta_* \in \bar{\delta}} \prod_i N(\delta_* g_i - \delta_* {g'}_i | \delta_* {g'}_i, \sigma_k) \tag{3.15}$$

   where,
   $\bar{\delta} \Longrightarrow$ set of partial derivate operators ,
   $\delta_* g' \Longrightarrow \sum_{\lambda \in D} t_D(\lambda) \delta_*(H_\lambda \mathrm{I})$,
   $\sigma_k \Longrightarrow \sqrt{2^{k+1}}\sigma_0$ as $k > 0$.

2. **Prior on the PSF:**
   Refraction causes any pixel to experience the set of translations which is represented by the PSF. The sparsity constraint on the PSF is applicable because UCWs are directional in nature. It is modeled by a Laplacian because the probability distribution of sparse images is heavy-tailed which is incorporated in the Laplacian. Thus by considering the PSF as an image, we get prior as

   $$p(t_D) = \prod_{\lambda \in D} \exp(-\beta t_D(\lambda)) \tag{3.16}$$

   where, $\beta \Longrightarrow$ rate of decay.

3. **Prior on the latent image:**
   The gradients of natural images are usually sparse. Although, a sparse prior will cause the latent image estimation problem non-convex [4]. Thus to avoid this we apply gaussian distribution for image gradients.

Energy minimization is used to estimate the latent image and the PSF for posterior probability. Energy function minimizes negative logarithm of posterior probability such as follows:

$$\begin{aligned} E(\mathrm{I}, t_D) &= -\log(\ p(\mathrm{I}, t_D|g)\ ) \\ &= -\log(\ p(g|\mathrm{I}, t_D)) \times p(\mathrm{I}) \times p(t_D)\ ) \\ &= -[\ \log(p(g|\mathrm{I}, t_D)) + \log(p(\mathrm{I})) + \log(p(t_D))] \end{aligned} \tag{3.17}$$

The above energy function is *non-convex*. Thus it is difficult to solve the problem. For the optimization procedure, they had used **alternating minimization (AM) framework**. They also proved the *bi-convexity* of the problem using AM framework. The basic idea of AM framework is that we fix one variable and minimizes the energy function across the other variable and then vice-versa. This will continue to happen iteratively until convergence or any predefined maximum number of iterations. Here we have to estimate the latent image and the PSF, so first we fix the latent image and estimate the PSF and then we fix the PSF that we estimated and estimate the latent image and this will continue till convergence.

Following are the steps of AM framework:

- **Prediction step:**
  First, we have to make the prediction of the latent image estimate I. We can predict initially the given observed blurred image as an estimate. However, to remove noise and other irrelevant details we apply different filtering techniques like bilateral filtering and gradient magnitude thresholding.

- **The PSF Estimation step:**
  We fix the latent image estimate which we acquired from previous prediction step and minimizes the energy function over PSF estimate as:

  $$E(t_D) = \sum_{\delta_* \in \bar{\delta}, \alpha_* \in \bar{\alpha}} \alpha_* \left\| \sum_{\lambda \in D} t_D(\lambda) \delta_*(H_\lambda \mathrm{I}) - \delta_* g \right\|_2^2 + \beta \left\| t_D \right\|_1 \tag{3.18}$$

  where,
  $\bar{\alpha} \implies \{\alpha_0\ \alpha_1\ \alpha_2\}$ as α_0 is weights applied to latent image, α_1 and α_2 are its first order and second order derivatives.
  In above optimization equation, we use l1-regularization i.e. LASSO.

- **The Latent Image Estimation step:**
  we fix the PSF estimate from above step and minimizes the energy function over Latent Image estimate as:

$$E(\mathrm{I}) = \sum_{\delta_* \in \bar{\delta}, \alpha_* \in \bar{\alpha}} \alpha_* \left\| \sum_{\lambda \in D} t_D(\lambda)\delta_*(H_\lambda \mathrm{I}) - \delta_* g \right\|_2^2 + \gamma \left\| \bigtriangledown \mathrm{I} \right\|_2^2 \tag{3.19}$$

  where, $\|\bigtriangledown \mathrm{I}\|_2^2 \implies (\frac{d\mathrm{I}}{dx})^2 + (\frac{d\mathrm{I}}{dy})^2$.
  Here we use conjugate-gradient method to estimate the latent image.

3.3.5 Algorithm for restoring underwater image using motion blur model

Algorithm 1: Algorithm for Recovering a Distorted scene in UCW [4]

```
Input: (i) Initial PSF estimate. (ii) Single observed distorted image(due to motion blur).
Output: Undistorted image(without skew and motion blur).
repeat
    From prediction step, compute the estimate of latent image $\hat{I}$.
    Calculate the PSF estimate by minimizing equation 3.18 using the given initial PSF
      estimate and the estimate of the latent image $\hat{I}$ which is obtained from the previous step.
    Then compute the estimate of the latent image $I$ by minimizing equation 3.19 by placing
      the result of the previous step as the estimate of PSF.
until (l-2 difference between latent image and its estimate is less than a threshold) OR
  (number of iteration does not reach its maximum value)
```

In paper [4], the value of the threshold is taken as 10^{-6}.

3.3.6 Limitations

There are some limitations to approach of using motion blur model which are as follows:

- This approach fails when there is a mixture of sine waves instead of a single sine wave because skewing effect becomes spatially varying.
- Also illumination changes affect the motion blur which in turns affect to our motion blur model.
- The blur may not be significant if the camera frame rate is high.

Chapter 4

Circular Ripples

There are many scenarios where circular wave pattern i.e. circular ripple is observed frequently. For example, one observes circular ripples in water bodies due to falling of rain droplets or pebbles into the water. A similar scenario occurs when water droplets decline from leaves of the trees or any other source to water puddles or water pool. Another scenario is children playing with pebbles and throwing them onto water surface to compete to see who's pebble covers more distance. All the above situation cause circular ripples. Due to refraction caused on the water surface, the image capture i.e. underlying planar scene will get distorted. Thus it is crucial to look into how to restore the distorted image in the presence of circular ripple.

4.0.1 Analytical Model for a single circular ripple

For generating circular ripples, the analytical model [4] is as follows:

$$h(x,y,t) = h_0 + A\sin\left(\sqrt{(x'w_x)^2 + (y'w_y)^2} - \phi(t)\right) \tag{4.1}$$

where,
$x' \implies (x - x_0),\ y' \implies (y - y_0)$,
$(x_0, y_0) \implies$ point on the water surface where ripple is originated.
Here we assume insignificant attenuation and small ripple fluctuations.

4.1 Challenges in Circular Ripple

Blur induced in the circular ripple is **space-variant**. We can prove this as follows:
Suppose f is the original observed image and g is the blurred image. The relation between f and g can be expressed as :

$$g = Hf \tag{4.2}$$

where, $H \implies$ space-variant blurring matrix. Here i^{th} column of H contains the blur kernel correlate to the i^{th} entry in f.
The transformation applied on the image plane at a particular time instance t is computed by differentiating the equation 4.1 :

$$w(\mathrm{x}, t) = h_0(1 - \frac{1}{n})\frac{Aw\cos{(w\sqrt{x'^2 + y'^2} - \phi(t))}}{\sqrt{x'^2 + y'^2}}\begin{bmatrix} x' \\ y' \end{bmatrix} \tag{4.3}$$

Here we can see that based on location every pixel will possess some translations, which produces geometric distortions on the image plane.

4.1.1 Solution: Polar coordinate system

We can express the equation 4.3 in polar coordinate system as follows:

$$w(\mathrm{x}, t) = h_0(1 - \frac{1}{n})Aw\cos{(wr - \phi(t))}\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{4.4}$$

where,
$r \implies \sqrt{x'^2 + y'^2}$,
$x' \implies r\cos\theta$ and $y' \implies r\sin\theta$

In the paper [4] they proved the claim that given A and w are fixed, the set of transformation experienced by pixels along the radial direction θ will be same. It can be explained by the equation 4.4. Suppose A, w, and θ are fixed, then the transformation at every pixel is dependent on the $\cos{(wr - \phi(t))}$ term. Suppose two pixels P_1 and P_2 have radial distances r and $r + \delta r$ along the direction θ. They undergo transformations which are function of $\cos{(wr - \phi(t))}$ and $\cos{(w(r + \delta r) - \phi(t))}$, it can also expressed as a phase shifted version of $\cos{(wr - \phi(t))}$ as new phase is $(w\delta r - \phi(t))$. This implies given long exposure, both pixels experience the complete cycle of the wave. Therefore blur caused by circular ripple is invariant in only radial direction. They also conducted an experiment where they show the PSF estimate of the image which is in cartesian coordinate and the PSF estimate of the image which is in the polar coordinate, as it is shown in the figure4.2.

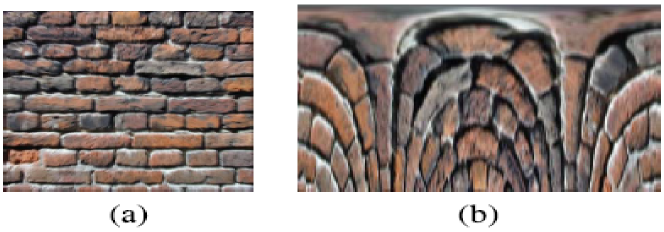

Figure 4.1: "(a) Brickwall in Cartesian coordinates, (b) image in polar domain(from Figure 9 [4], reproduced without permission)"

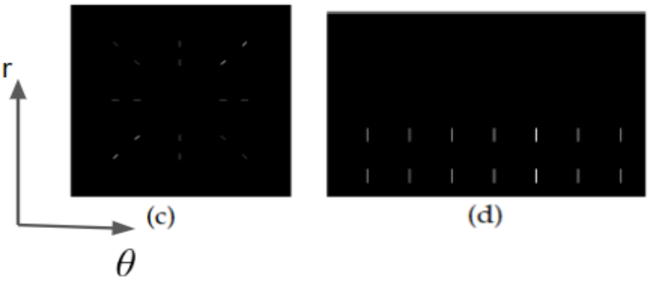

Figure 4.2: "(c) estimated PSFs at various locations in (a), (d) estimated PSFs at different locations of (b)(from Figure 9 [4], reproduced without permission)"

The figure 4.2 shows that vertical line corresponds to θ in polar coordinate maps to all the pixel which lie in the radial direction θ in the Cartesian coordinate system.

Therefore for the restoration of the image, we convert the observed blurred image into the polar domain as blur is space-invariant in the radial direction. Then we call the AM framework which is used to restore the images distorted by UCW. The image is then restored, and the restored image is converted from the polar domain back into the Cartesian domain. This is explained by the following algorithm :

Algorithm 2: Algorithm for Recovering a Planar Scene Distorted by circular ripple [4]

```
Input: (i) Initial PSF estimate in polar domain. (ii)Single observed distorted image(due to
       motion blur) in the presence of circular ripple.
Output: Undistorted image(without skew and motion blur).
Due to issue of space-variant, convert the observed image from cartesian domain to polar
  domain.
Then use the result of the previous step as an input to algorithm 1 who estimate the original
  image which is in the polar domain.
Revert the resultant original image back to the Cartesian coordinate system. .
```

4.2 Limitations

Since the basic principles of this method are similar to those presented for the UCW model3.3, the limitations are also quite similar to the UCW model. These are listed here below again, for the sake of completeness.

- This approach is valid only in the case of single ripple and does not consider the case when the mixture of circular ripples occurs.
- Because of we model our problem in terms of motion blur it will be susceptible to illumination changes.

4.3 Experimental Results of Circular Ripples

I have simulated the circular ripple from the analytical model explained earlier. Where values of w, A were tweaked to get the water wave-like simulation which looks as good as conducting a real experiment. For the optimum result we have used $h_0 = 20$, A i.e. amplitude $= 0.5$ and wavelength $= 300$ and choice of $w = \frac{2\pi}{\text{wavelength}} * (\text{some constant})$ here we choose constant to be $\cos\theta$ as $\theta = \frac{\pi}{4}$. The following figure shows the quiver plot of a single circular ripple originating from the center of the figure. The quiver plot represents the resultant direction of the gradient vector at each pixel in the image.

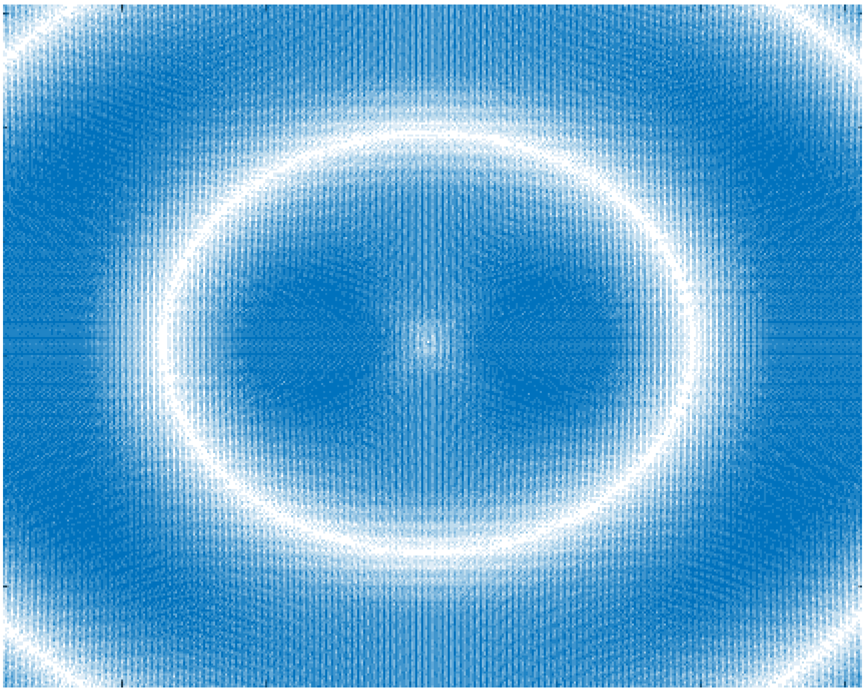

Figure 4.3: Quiver Plot of Circular Ripple

Here is the link where quiver plot is shown in higher resolution (stored in .eps and .fig format): https://drive.google.com/open?id=14qjGI_4qym1IcDgJkRC9NkJwTqWzsG3m

Here is the link to the video of the simulation of single circular ripple where the scene is submerged beneath the water surface : https://drive.google.com/open?id=1rRoD__mqCYvmWweBYSXXHaWEo_rtzF5Z

Also for the experimental purpose, we tried to convert the average displacement matrix of the image which is formed using equation 4.3 to grayscale image at the particular frame. In the figure 4.5, darker shade means value of average displacement gradient experience by that pixel is negative and lighter shade represents value is positive.

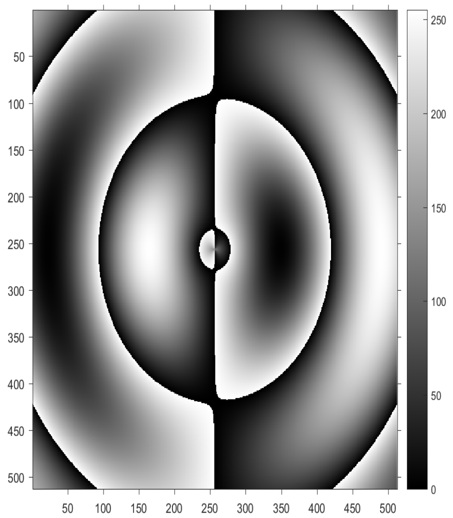

Figure 4.4: Displacement at 600 frames

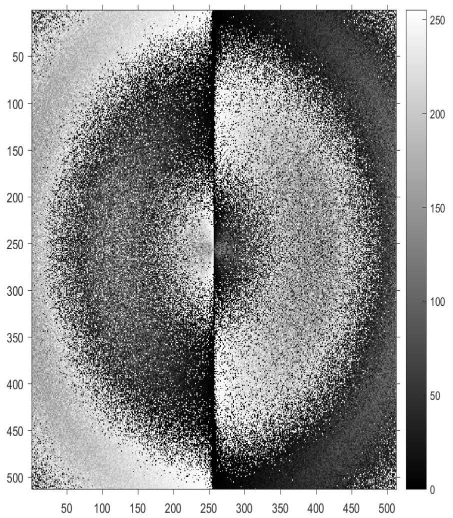

Figure 4.5: Displacement at 900 frames

Here we can see in the figure circular pattern are forming to emulate circular ripple. We empirically verified that average displacement in case of single circular ripple is approximately close to zero.

To overcome the limitations of method mentioned in section 4.2, we try the approach used in section 3.1. However, instead of tracing trajectories of all the points we attempt to trace trajectories of some salient feature point which we get by implementing any feature extraction algorithms like SIFT features or Harris corner detector. While conducting experiment we apply some threshold to choose the feature point. After choosing feature points, we can use any feature tracking algorithms. Here we use **Kanade–Lucas–Tomasi (KLT) feature tracking algorithm**. Our aim is to derive the model parameters by employing regression algorithms on the tracked feature points. Given the model parameters, we would infer the shape of the water surface, and this would help to restore the underlying scene. We implemented the following procedure for tracking the points:

- Identify the salient feature points using a SIFT features for saliency detection.
- Then applying some threshold to pick only good feature points. This is done to reduce the computation of KLT tracker algorithm.
- Apply KLT tracker to track the feature point which we get from the previous step.

Following figure show the tracking results of the image of the pocket watch. For simplicity, we have shown the tracking of a single salient point.

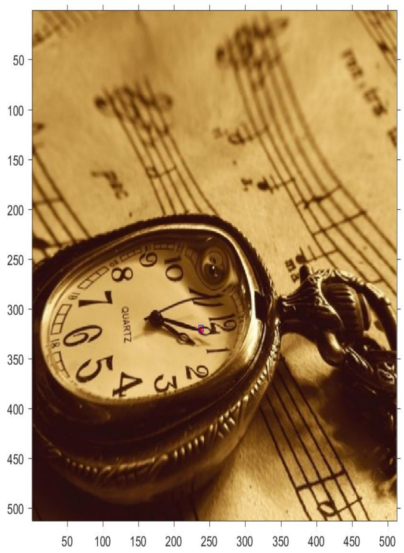

Figure 4.6: Tracking Result at Initial frame

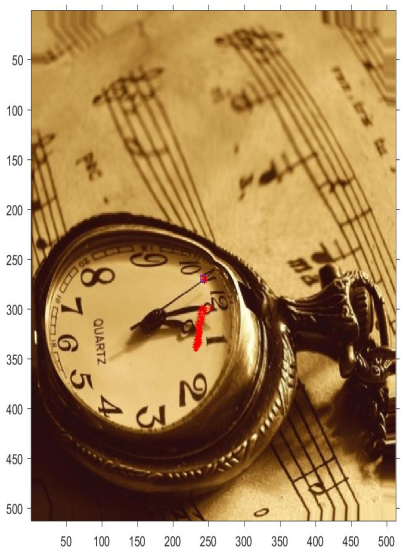

Figure 4.7: Tracking Result After 60 frames

As we can see in figure 4.7 our method fails to track salient feature point after 60 frames. Here we are tracking the end of minute hand but at 60 frame our tracker is tracking point on second hand at clock time 11. We tried different images but our method fails to track salient feature point after some frames. Therefore we can see that traditional feature tracking methods do not work in our circular ripple scenario.

4.4 The Mixture of Multiple Circular Ripples

As we see that the previous approach does not work with multiple ripples and there are scenarios where the mixture of multiple ripples occurs more often than one thinks such as while rainfall, ripples generated by rain droplets are originating more closely which leads to generating a pattern that is quite similar to pattern generated by mixture of multiple ripples.

4.4.1 Analytical Model

For simulating the mixture of multiple ripples, analytical model from section 4.0.1 can be extended:

$$h(x, y, t) = h_0 + \sum_{i=1}^{n} A_i \sin\left(\sqrt{(x_i' w_{x_i})^2 + (y_i' w_{y_i})^2} - \phi_i(t)\right) \tag{4.5}$$

where, $i \implies$ index of ripple and $n \implies$ the Total number of ripples.
Other parameters are same as the previous equation 4.1 only difference is that each ripple has its own amplitude and different originating points.

4.4.2 Experimental Results

We also simulate the mixture of two circular ripples where we kept the parameter such as wavelength and amplitude same only origin of two ripples are different. By simulating this we got quiver plot as shown in the following figure 4.8.

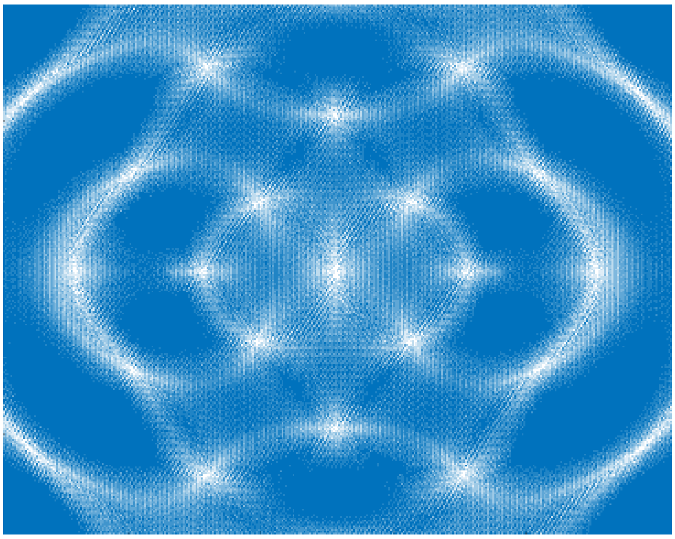

Figure 4.8: Quiver Plot of the Mixture of Two Circular Ripples

Here in figure 4.8 we can clearly see that two ripples interfere with each other and create the odd pattern.

Here is the link to the video of the simulation of the mixture of the multiple circular ripple where the scene is submerged beneath the water surface : https://drive.google.com/open?id=1Q7ImR2fHoF1rIEd7762eI9-gOSVbBgW9

Chapter 5

Proposed Work

The primary focus of my research work will be on the restoration of the scene degraded by circular ripple. For this purpose, I am planning to do the following things:

- To try the deep learning techniques for salient feature point matching throughout the video. (As the traditional tracking approach fails to solve the problem.) Then we can use the trajectories which were obtained through point tracking, to determine the parameters of the circular wave.
- To extend the approach in which we succeed in finding a solution for a single circular ripple to find a solution for the mixture of multiple circular ripples.

Chapter 6

Conclusion

This report describes the problem of image restoration of underwater images. The report proves analytically that distortion function is proportional to the height of the water surface and the effect caused due to refraction is space-variant.

The *reconstruction of water surface* method, simultaneously recover underlying scene and reconstruct the water surface using optical flow method like tracing the trajectory of each point, then computing centre of the trajectory and afterwards calculating gradients then by integrating. It is generalized so that it can be applied to a variety of cases. However, it is computationally expensive to perform tracking of all points on the image.

The *model-based tracking* method's outcome is same as the above approach but it executes different procedure such as it uses PCA which one of the dimensionality reduction methods to generate bases called "Water bases" which encapsulates the behaviour of water wave. Then it uses no template model by implementing some heuristics to estimate the coefficients of the water bases. However, the choice of the number of PCA bases is a free parameter and if choose badly can affect the outcome of the method. Also to get accurate results higher frame rate is required. Both this approaches do not consider a case of motion blur if exposure was long.

The *motion blur* model, first it was shown that information about skew is incorporated in motion blur model. This resultant blur model is space-invariant. It uses MAP formulation to find the latent image and warping function given a blurred observation. It uses the method of alternating minimization(AM) to convert the function into bi-convex. Due to blur model, it will be affected by changes in illumination.

This report shows the mathematical expression for simulating circular ripple. It uses motion blur model to reconstruct the distorted image. But first, it converts the domain from cartesian to polar as in polar domain effect is space-invariant. However, this approach does not work in the mixture of multiple ripples. To try the different approach one has to know the water surface parameter which can be obtained by tracking the points. But instead of tracking all the points as given in this approach described in section 3.1 it is advisable to track some salient feature points to reduce the computational overhead. As shown in the report traditional tracking method like KLT failed to track the salient points through all the video sequences. So to track feature points report purpose the machine learning techniques.

Chapter 7

References

[1] ROBERT T. FRANKOT and RAMA CHELLAPPA. A method for enforcing integrability in shape from shading algorithms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 10, 1988.

[2] Pietro Perona Marina Alterman, Yoav Y. Schechner and Joseph Shamir. Detecting motion through dynamic refraction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 35, 2013.

[3] Hiroshi Murase. Surface shape reconstruction of a nonrigid transparent object using refraction and motion. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages Vol. 14, No. 12, 1992.

[4] Karthik Seemakurthy and Ambasamudram Narayanan Rajagopalan. Deskewing of underwater images. In *IEEE TRANSACTIONS ON IMAGE PROCESSING*, volume 24. IEEE, 2015.

[5] Y. Tian and S. G. Narasimhan. The relationship between water depth and distortion. In *CMU TechReport RI*, 2009.

[6] Y. Tian and S. G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Proc. IEEE 12th Int. Conf. Computer Vision*, pages 2303–2310. IEEE, 2009.

[7] Christina Zdanowicz. How to shoot amazing video from drones. `https://edition.cnn.com/2014/05/22/tech/innovation/drone-uav-photography/index.html`, 2014. [Online website article updated 22-May-2014].