Adaptive Filters to remove Blurring Effects over Time for Underwater Surveillance

Konstantinos Vougioukas

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2012

Abstract

Exploring the ocean bottom has always been an area of great scientific and environmental concern. However, study of the underwater environment up until recently was very difficult due to the extreme conditions. With the advances in underwater photographic equipment surveillance of the sea bed is now easily realizable. The quality, however, of underwater images is still worse than that of images shot in the air and images usually appear hazy. This thesis deals with the problem of underwater surveillance of a scene. The quality of the recording obtained by the camera deteriorates over time due to problems like dirt/water on the lens and the glass protecting the camera, which is why the camera must be cleaned regularly. The dirt on the lens as well as floating particles create a blur and noise in the frames of the video. This projects' main goal is to remove the blur effects from the underwater videos. As a secondary goal we wish to develop a method that uses the temporal information of the video as well as the knowledge of when the camera was cleaned. The method proposed in this study solves the problem in two stages. It first removes any noise that is present in the recordings and then deals with the blur effects. For the denoising stage a variation of the BM3D algorithm [8] was developed. Several different approaches were implemented for the deblurring problem based on the multiframe blind deconvolution method described in [1]. Evaluation of the algorithms was held for both artificial and real degradation of the frames.

Acknowledgements

Many thanks to my supervisor Bastiaan Boom for his constant support and excellent advice throughout this project. I greatly appreciate his patience and his willingness to help me understand in depth the highly technical, mathematical aspects of this thesis. I would also like to thank my brother for his support during this masters.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Konstantinos Vougioukas*)

To my family.

Table of Contents

1 Introduction 1

2 Background 4

2.1 Degradation Models 4

2.2 Denoising Methods 6

2.2.1 Spacial Denoising Methods 6

2.2.2 Temporal Denoising Methods 12

2.3 Deblurring Methods 12

2.3.1 Deconvolution Methods 13

2.3.2 Blind Deconvolution Methods 16

3 Methodology 18

3.1 General Algorithm Description 18

3.2 Background Subtraction 19

3.3 Denoising Step 22

3.3.1 Grouping Stage 23

3.3.2 The Collaborative Filtering Stage 24

3.3.3 Aggregation Stage 25

3.4 Deblurring Step 26

3.4.1 Method Description 26

3.4.2 Method Variations 28

4 Experiments and Evaluation 31

4.1 Methods of Evaluation 31

4.2 Denoising Experiments 32

4.2.1 Experiment with Additive Gaussian Noise 34

4.2.2 Experiments with Salt and Pepper Noise 39

4.3 Deblurring experiments 41

4.3.1 Performance on Artificial Global Blur 41
4.3.2 Experiments with Artificial Local Blur 45
4.4 Experiments with Real Data 47

5 Conclusions and Discussion 51
5.1 Conclusions 51
5.2 Future Work 52

Bibliography 53

Chapter 1

Introduction

Since the invention of the first "amphibious" camera in 1960 the interest in underwater videography for ecological and recreational reasons has greatly increased. Despite the technological advances in the equipment the quality of underwater images and videos is still much worse than that of images shot in the air because of the limitations imposed by the physical properties of the water medium. Underwater scenes are characterized by their poor visibility due to the fact that as light travels deeper into the water it gets exponentially attenuated. This results in images and videos that are hazy, dark and have bad contrast.

In surveillance systems there is the added deterioration of the recordings over time, due to dirt/water build up on the lens and the glass protecting the cameras. The deterioration is far worse in the case of underwater surveillance because of the vast amount of dirt and floating particles present in the water.

Underwater image processing has received considerable attention over the last few decades due to its challenging nature and its importance for the environment. Improving the underwater image quality can be separated into two different problems known as the *image restoration problem* and the *image enhancement problem*.

Image restoration aims at estimating the true scene by removing the noise and inverting the degradation process. Doing this usually requires building mathematical models of the degradation and using various signal processing filtering techniques. Classical image restoration methods are Wiener filtering and blind image deconvolution. An example of the results of image restoration is shown in Figure 1.1.

On the other hand image enhancement aims at making the images more aesthetically pleasing through subjective criteria and without relying on complex mathematical models. Colour correction, contrast and brightness adjustment are good examples of

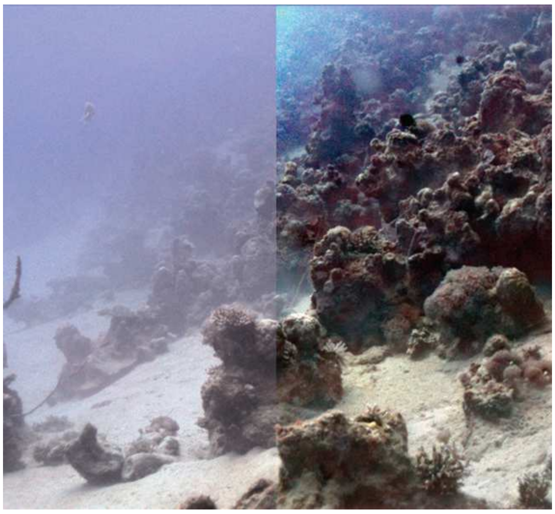


Figure 1.1: Example of underwater image restoration [2].

image enhancement methods. An example of the results of colour correction can be seen in Figure 1.2.

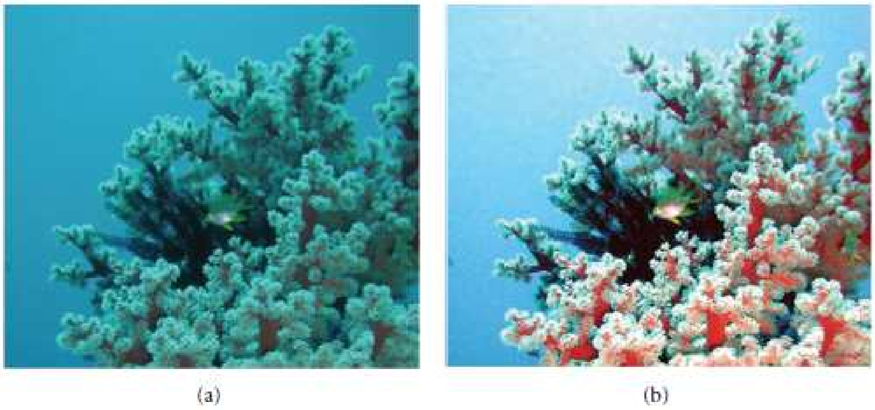


Figure 1.2: Example of colour correction of an underwater image. a) original image b) colour corrected image[2].

The purpose of this research is to restore the video recorded by an underwater surveillance camera back to its original quality using variations of state of the art methods. Focus will be laid on dealing with the video restoration problem and not on video enhancement. The data consists of video sequences whose quality deteriorates with time as more dirt gathers on the lens. This deterioration can be observed in Fig. 1.3.

The deterioration of the images can be split into two different types. The first is a local blurring of the image in places where there is dirt. This blur can't be considered stationary throughout the sequence as it sometimes tends to shift slightly back and forth depending on the the water currents. The second is noise that is present from

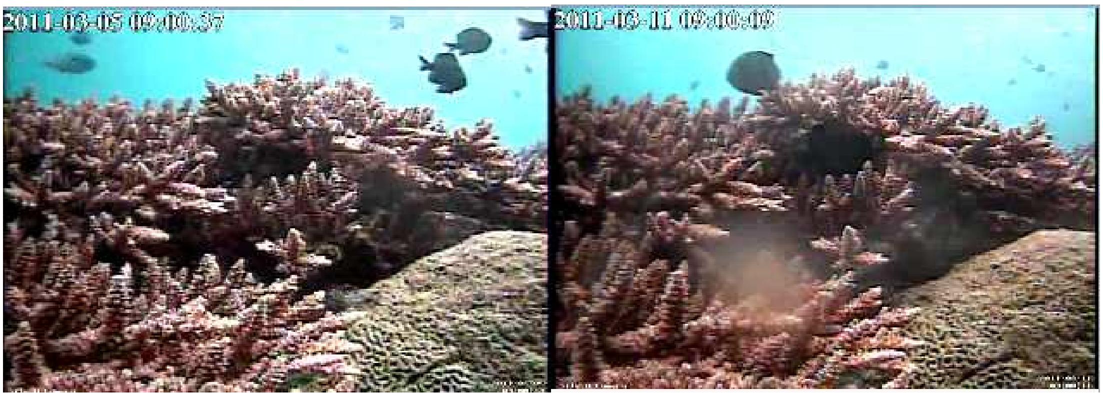


Figure 1.3: Frames taken a) with recently cleaned lens b) lens with particles and dirt on.

either floating particles or camera measurement noise (errors in the analog-to-digital conversion or during the quantization). In order to maintain the video quality at a standard that allows for the monitoring of underwater environment the lens must be cleaned in regular intervals. This procedure is costly and the frequency with which it is performed could be reduced if the image is restored using image restoration techniques.

Some video restoration techniques deal with each frame in a video sequence separately thus ignoring the temporal relationship between consecutive frames. In the surveillance problem, where the camera is stationary and the scene doesn't change significantly from one instant to the other past frames hold valuable information. This project aims to develop a denoising and deblurring method for the surveillance problem that makes use of this information. In order to further exploit this information an attempt is made to utilize the frames recorded when the camera lens has been recently cleaned. These frames are easy to detect since knowledge of when the camera is cleaned is available.

Chapter 2

Background

In this chapter the video restoration problem is more strictly defined. It is divided into the two separate problems of video denoising and video deblurring. These two problems are mathematically formulated and popular methods for solving them are analysed and discussed.

2.1 Degradation Models

The purpose of image or video restoration is to reverse any defects that alter an image or frame. There are many forms of deterioration that can be modelled in different ways. If an assumption is made that the original frame is corrupted only by additive noise as seen in Figure 2.1 then we have what is known as the denoising problem. This problem is described by the equation (2.1), where $f(x,y,t)$ is the original frame, $\eta(x,y,t)$ is the noise term and $g(x,y,t)$ the measured frame.

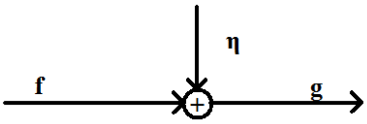
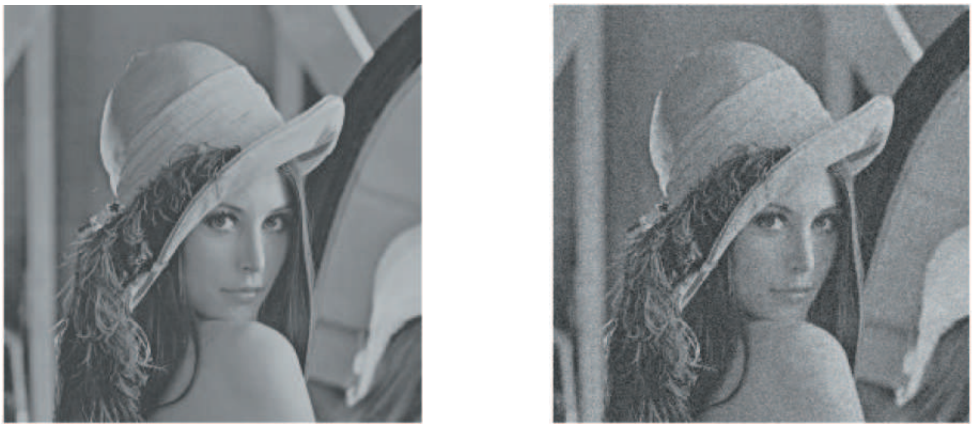


Figure 2.1: Degradation model assumed in the simple case of the denoising problem.

$$g(x,y,t) = f(x,y,t) + \eta(x,y,t) \tag{2.1}$$

The effect this sort of degradation has to an image is shown in Figure 2.2. As we can see noise appears as randomly-spaced speckles in an image. Noise can be caused by various reasons such as quantization errors, compression errors or high camera ISO sensitivity.



(i) Original image

(ii) Noisy image

Figure 2.2: Example of an image being corrupted by additive noise

In cases where blur is also present we have the deblurring problem shown in Figure 2.3 and described by (2.2) for each time instant.

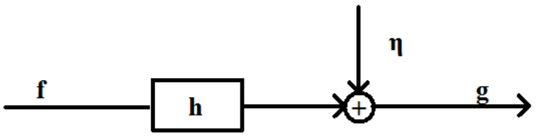


Figure 2.3: Block diagram for the degradation model assumed in the case of the deblurring problem.

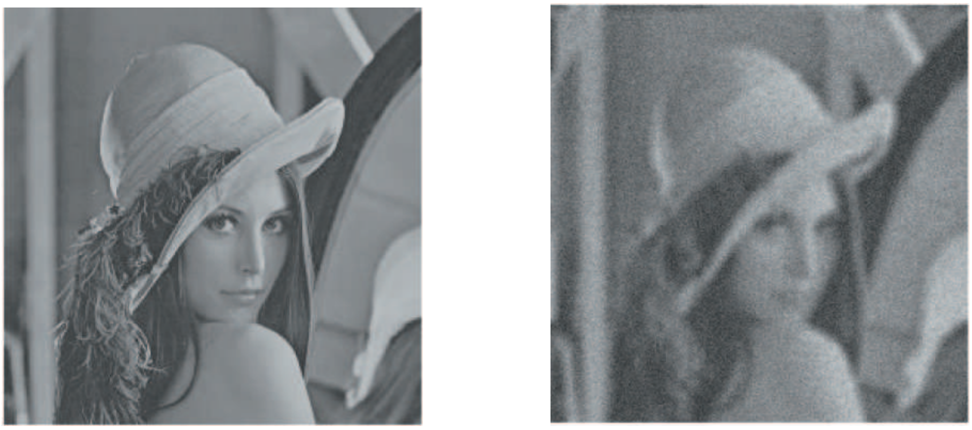
$$g_t(x,y) = h_t(x,y) \star f_t(x,y) + \eta_t(x,y), \tag{2.2}$$

where $h_t(x,y)$ is the degradation point spread function at time instant t and $\star$ is the convolution operation

$$f \star h = \sum_{(n,m)} f(x,y)h(x-m,y-n).$$

In this case the corrupted image is obtained by passing the original image through a blurring system and then adding noise to it. An example of the sort of degradation this causes to the image is shown in Figure 2.4.

In this problem it is evident that a lot more details of the image are lost compared to the case of just additive noise.



(i) Original image

(ii) Degraded image

Figure 2.4: Example of image degradation in the problem assumed in image deblurring.

2.2 Denoising Methods

Video denoising methods can be split up into spacial(section 2.2.1) and temporal methods (section 2.2.2) based on whether they use the temporal relationship between frames in the video sequence or not. Although temporal methods make better use of this information provided by videos they are usually more complex and require motion compensation in order to avoid artifacts created when blending together pixels from different frames.

2.2.1 Spacial Denoising Methods

Spacial denoising video methods are image denoising methods applied on each frame separately. Typical ways of solving the denoising problem are to apply linear or nonlinear filtering to the image. This filtering can take place in both the space or frequency domain.

2.2.1.1 Low-pass Filtering

The most common type of filtering used in images is the linear low-pass filtering. The simplest linear filter is perhaps the mean filter. It is based on the assumption that adjacent pixels are likely to be similar to each other. It is implemented with the standard sliding window approach using a convolution mask. The result of this convolution is that each pixel in the image will be replaced by the average of its eight neighbours. A generalization of the mean filter is the space domain averaging filter, which does not need to weight all neighbours equally. Good examples of other spacial-domain averaging filters are the 5-point weighted averaging filter and the Gaussian filter. Some of the masks used in these filters are shown in Figure 2.5. The problem with these sorts of filters is that they tend to blur edges and details of the scene. This problem gets worse as the size of the convolution window increases.

1/4	1/4
1/4	1/4

(i) 2×2 convolution mask used in mean filtering

0	1/8	0
1/8	1/4	1/8
0	1/8	0

(ii) 5-point weight averaging

Figure 2.5: Examples of convolution masks of space domain averaging filters

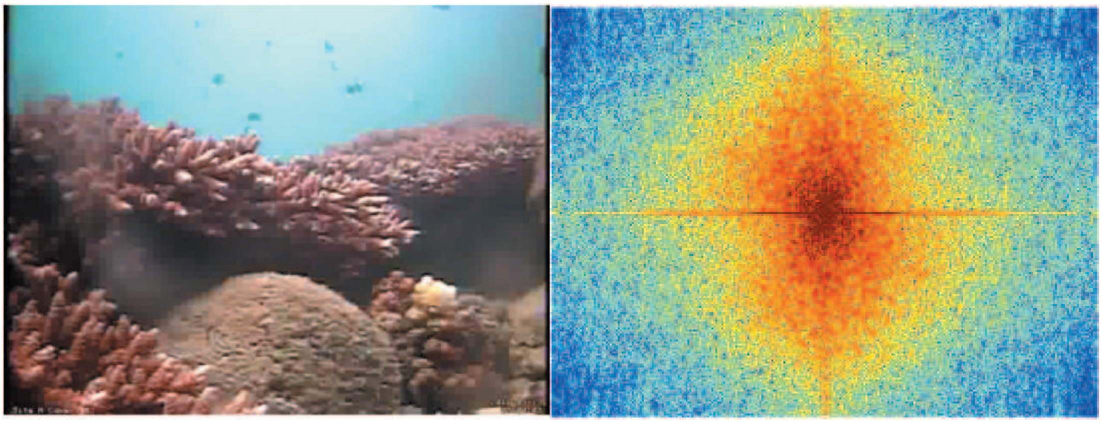
It is easier to understand how these low-pass filters work by looking at the frequency components of the images. Frequencies in images correspond to the rate of change in pixel intensities across an image. Low frequencies correspond to the large features of an image (e.g. homogeneous regions) whereas high frequencies correspond to rapid pixel changes that occur in an image (e.g. noise, edges). Therefore the original image will usually have more energy in the low frequencies than in the high ones, whereas the noise will have more energy in the high frequencies. This property can be easily verified by looking at the frequency representation of a noiseless image taken from our underwater recordings (Figure 2.6ii).

We can see that the red areas, which correspond to the high energy content are situated in the centre of the plot where the low frequencies are. It is also obvious that there are still few high frequencies that have a significant energy content.

Applying the fast fourier transform (FFT) to the degraded image $g(x,y)$ will produce the frequency representation of that image $G(u,v)$. Filtering can then be carried out by multiplying this with the frequency response of a low-pass filter (Butterworth, Gaussian, etc.) like the one shown in Figure 2.7 which greatly attenuates the high frequency coefficients. This multiplication in the frequency domain corresponds to convolution in the spacial domain according to the well known property of the convolution theorem ($f \star g = FG$).

This produces an output $F(u,v)$ given by the following equation

$$F(u,v) = G(u,v)\hat{H}(u,v), \tag{2.3}$$



(i) Image taken from underwater recordings (ii) Frequency representation of the image

Figure 2.6: A clean image along with its frequency representation. The red areas denote high energy content and the blue areas denote low energy content. Frequencies increase as we go from the middle of the plot to the edges.

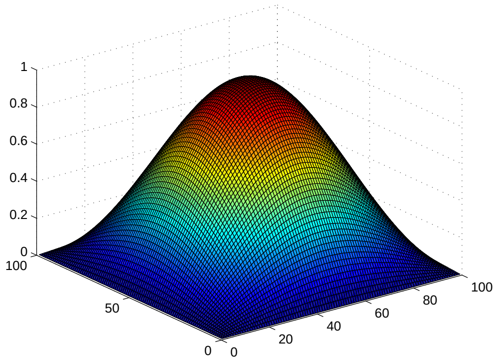
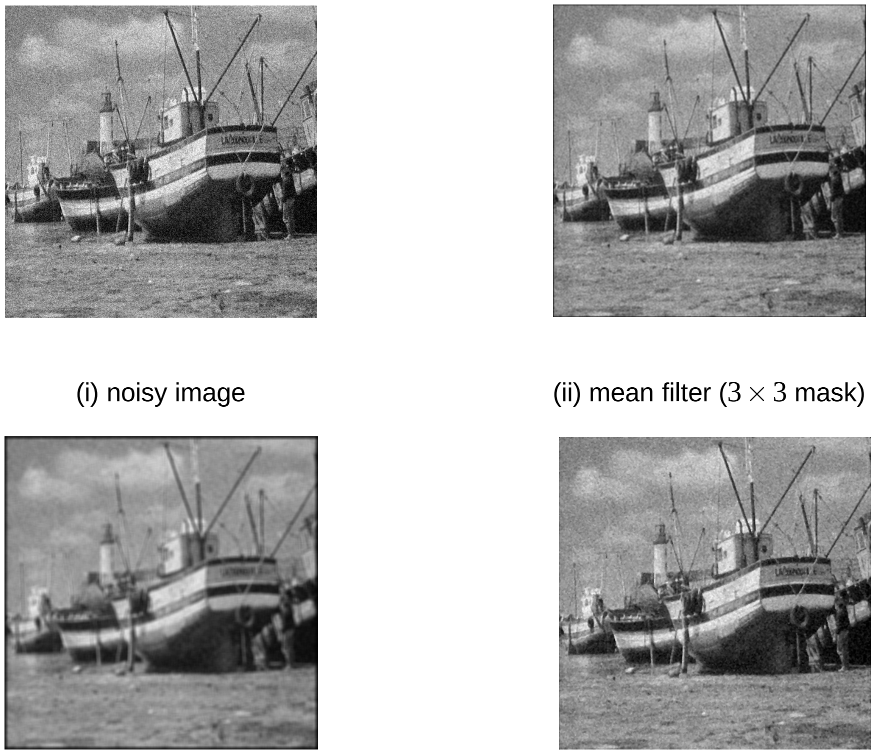


Figure 2.7: Frequency response of a Gaussian low-pass filter.

where $\hat{H}(u,v)$ is the frequency response of the filter. Finally, the inverse linear transform is applied to return to the spacial domain. Applying low-pass filtering in the frequency domain gives a more intuitive approach due to the fact that it allows for visualization of the frequency components.

An non-linear alternative to the mean filter is the widely used median filter, which replaces each pixel in an image with the median of its surrounding pixels. This filter performs better than the averaging filters for salt and pepper type noise and does not suffer as much from blurring effects. It does however tend to perform poorly when the number of noise pixels consist of more than half of the window pixels. A more sophisticated version of this algorithm can be found in [3]. The results of mean and median filtering can be seen in Figure 2.8.



(i) noisy image

(ii) mean filter (3×3 mask)

(iii) mean filter (7×7 mask)

(iv) median filtering

Figure 2.8: Filter performance for an image corrupted with gaussian noise with zero mean and 0.05 variance

2.2.1.2 Patch Based Methods

All low-pass filtering techniques share a common problem that stems from their assumption that natural images have more information in the low-frequencies than in the high frequencies. Noise however will affect equally all coefficients. Thus removing the high frequency coefficients will eliminate more noise than signal but any information in the high frequencies is lost and any noise in the low frequencies persists.

A category of image denoising methods that is gaining popularity is that of patch based image restoration methods. These methods try to find similar patches within the image and use their spacial redundancy to denoise the image. This is based on the assumption that images will contain small patches that are quite similar due to the repetitive patterns or elongated edges in a scene. This assumption is usually valid for most natural scenes. The first patch-based method that was created was the *Non-local means* algorithm [4], which does not make the assumption of high frequency noise. Non-local means gets its name from the way that it operates. It assigns a window (patch) centred around each pixel in the image and measures the similarities between the patches. Each pixel is then estimated as a weighted average of all the other pixels in the image. The weighting of each pixel depends on the similarity of its patches to the reference patch, which is determined by a distance measure. This is better understood by the example shown in Figure 2.9.

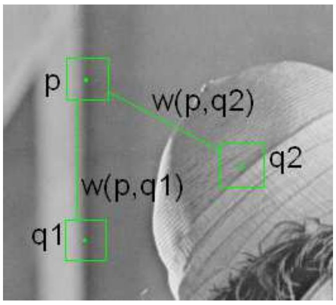


Figure 2.9: Example showing how patch similarity affects the weighting of the average. Weighting of q_1 will be larger than that of q_2 due to its patch's similarity with the reference patch. Image taken from [4]

In this example we want to estimate the value of the pixel p as a weighted average of other pixels in the image. It is safe to assume that NL-means will give a large weight $w(p,q_1)$ to pixel q_1 since it is evident by inspection that the patch centred around this pixel is very similar to the patch from the reference pixel p. In correspondence with this similarity rule, the pixel q_2 will probably be assigned a small weight $w(p,q_2)$. Therefore pixel p will end up taking a value that is much closer to q_1.

2.2.1.2.1 BM3D Denoising Method The latest development in patch based algorithms is the BM3D algorithm, which is currently considered the state of the art in image denoising yielding very impressive results. The BM3D method, much like the non-local means, uses of a distance measure to assess the similarity between two patches. BM3D has two denoising steps in order to ensure better noise reduction.

In the first step BM3D performs an exhaustive search on the image to find similar patches for every patch in the image. The patch size can vary and typical sizes are 8×8, 16×16 or 32×32. Once the similar patches have been determined they are grouped together to form blocks. Blocks can contain overlapping patches.

Next, collaborative filtering is performed on the blocks to produce estimates of the patches. The first stage in this collaborative filtering is to perform a linear transform on the block so as to get its frequency representation. This transform will obviously have to be a 3 dimensional transform since the blocks are 3D. This is followed by noise reduction via thresholding the transform coefficients. This is called hard-thresholding and is a special case of magnitude thresholding. Magnitude thresholding as its name implies compares the magnitude of the transform coefficients to a threshold and sets them to zero if they are less. It is based on the assumption that natural images are very likely to have only a small number of high frequency non-zero coefficients. Noise does not usually contain a lot of energy so it is expected that after the additive noise these high frequency coefficients will still be rather small. Setting these small coefficients back to zero will eliminate some high frequency components with very little signal information but maintain high frequencies that correspond to edges. There is ,however, a trade-off since components that contain both noise and signal will not be affected by the thresholding some noise will still be present in the image. The inverse linear transform is then applied to obtain the estimates of the blocks. Some representative techniques that use magnitude thresholding can be found in [5], [6], [7].

When the collaborative filtering is finished, we get an estimate for each patch and a number of estimates for each pixel (due to the fact that the pixel may be present in more than one patch). In order to have a single estimate of the pixel, an aggregation of the estimates is performed. This is a weighted average where each pixel estimate is weighted according to the number of maintained coefficients in its block after the hard thresholding. This concludes the first denoising step.

The second step is basically a modified repetition of the first step, which uses as inputs the previous block estimates and creates blocks based on them. Similarly to the first step, collaborative filtering is performed on the blocks only this time the filtering is done using a Wiener filter instead of hard-thresholding. The group estimates are obtained by performing the inverse transform and then final estimates of the patches are made using aggregation. As mentioned earlier the purpose of the second step is to further denoise the basic patch estimates. The whole procedure followed by the algorithm can be seen in Fig.2.10.

Patch-based algorithms generally perform better than other methods and have the added advantage of being easily transformed into temporal video denoising methods. However, the exhaustive search they perform to identify similar patches usually makes them far slower and worse for on-line applications.

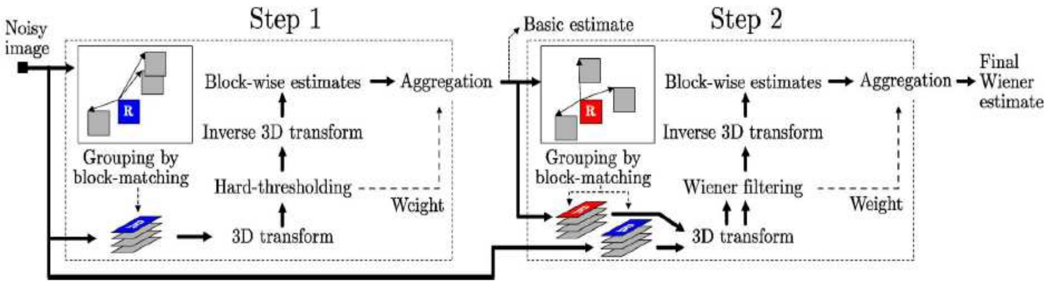


Figure 2.10: Flowchart of the BM3D algorithm showing both steps that lead to obtaining noiseless images.[8]

2.2.2 Temporal Denoising Methods

Temporal denoising methods are better suited for video denoising because they make use of the fact that scenes will not change too much from frame to frame but the noise will probably not be present in the same pixels. Before the discovery of patch-based methods temporal denoising methods consisted of very complex algorithms using wavelet filtering and some sort of motion estimation to compensate for any motion blur from moving objects in the scene [9]. Their complexity along with their unimpressive results made them unappealing and as a result applications approached the denoising of video as a series of independent single frame denoising problems. After the introduction of NL-means, the potential of patch-based methods for temporal denoising was made evident. Most of the temporal patch-based methods perform searches both in space and time to find similar patches and perform patch-based restoration using these similarities. This is better understood by looking at Figure 2.11.

The figure illustrates how similar patches (green squares) can be found in different frames of the sequence as well as in the same frame. Due to the nature of patch based methods the only thing that has to be modified in order for them to work using multiple frames is the search for similar patches. The filtering of the patches can be the same as the one done in spacial patch-based methods.

2.3 Deblurring Methods

Blur is a very common problem in video sequences that can be caused by a variety of different reasons such as moving objects, unstable cameras or even due to the atmosphere. Blurry images can be seen as a result of applying a filter with similar properties to those of a low-pass filter on the original frames $f_t(x,y)$ of the video. The impulse

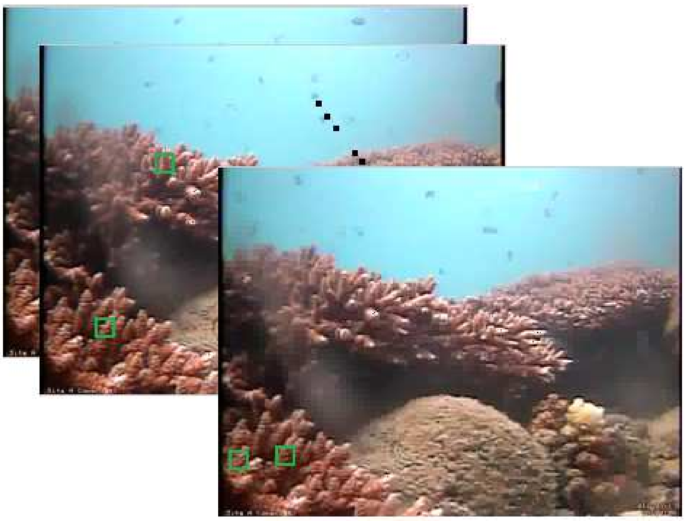


Figure 2.11: Example showing how temporal patch-based methods find similarities between patches belonging to different frames.

response of this filter h(x,y) is also known as the point spread function (PSF) of the blur. The output frames $g_t(x,y)$ are calculated by doing

$$g_t(x,y) = h_t(x,y) \star f_t(x,y)$$

or equivalently in the frequency domain by

$$G_t(u,v) = H_t(u,v)F_t(u,v),$$

Where $G_t(u,v)$, $H_t(u,v)$, $F_t(u,v)$ are the 2D fourier transforms of g, h and f respectively.

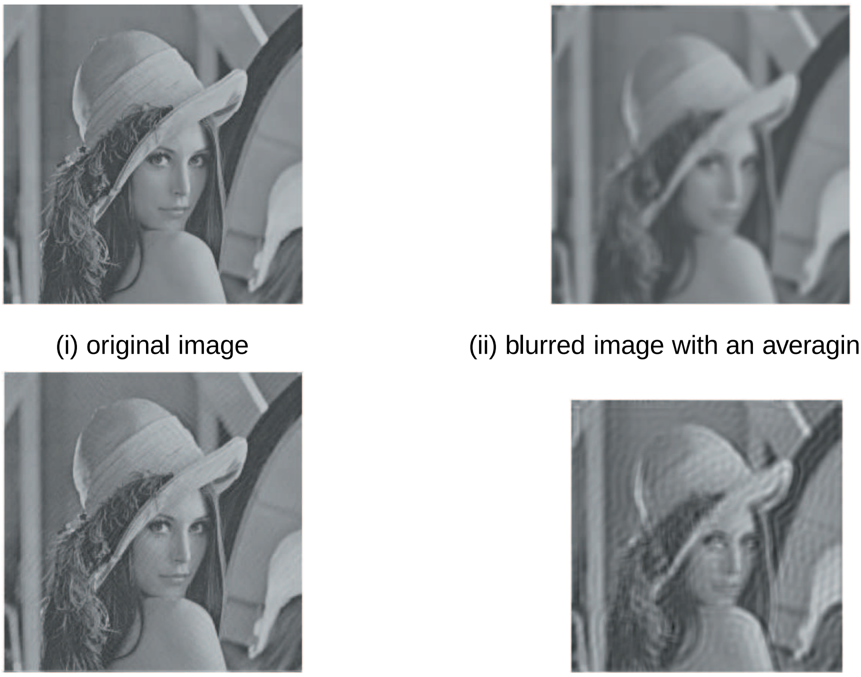
Deblurring is in fact an inverse problem, which aims at reversing the effects of the blur PSF. If the PSF is known then the problem can be solved using what are known as *image deconvolution* techniques whereas if the PSF is unknown then *blind-image deconvolution* must be applied.

2.3.1 Deconvolution Methods

The quickest and most naive way to do non-blind image deconvolution is called inverse filtering. Inverse filtering calculates the original scene in the frequency domain as

$$F(u,v) = \frac{G(u,v)}{H(u,v)}$$

One can immediately detect a problem here as $H(u,v)$ becomes very small $\frac{1}{H(u,v)}$ approaches infinity. This can be avoided by not allowing $H(u,v)$ to become smaller than a threshold, although if too many coefficients of H are lost then the image appears distorted. This effect can be seen in Figure 2.12.



(i) original image

(ii) blurred image with an averaging filter

(iii) Deblurred with inverse filter with threshold at 0.05

(iv) Deblurred with inverse filter with threshold at 0.3

Figure 2.12: The effect the lower bound has on the reconstruction of inverse filtering.

As we can see if the lower bound is small enough the reconstruction is quite accurate. However there is also another more significant problem associated with this method. Inverse filtering is basically a form of high-pass filter, making it very sensitive to additive noise which is almost always present in frames. This issue can be seen in Figure 2.13 and renders inverse filtering virtually useless for real applications. A solution to this problem is presented by Wiener filter deconvolution [10]. Wiener filtering finds an optimal compromise between inverse filtering and denoising. The Wiener filtering is a linear estimator based on the orthogonality principle, which makes it the optimal linear estimator with respect to the least squares error. The frequency response of the Wiener filter is shown below

$$W(u,v) = \frac{H^*(u,v)\,S_{gg}(u,v)}{\|H(u,v)\|^2\,S_{gg}(u,v) + S_{\eta\eta}(u,v)}, \tag{2.4}$$

(i) Blurred image with additive noise

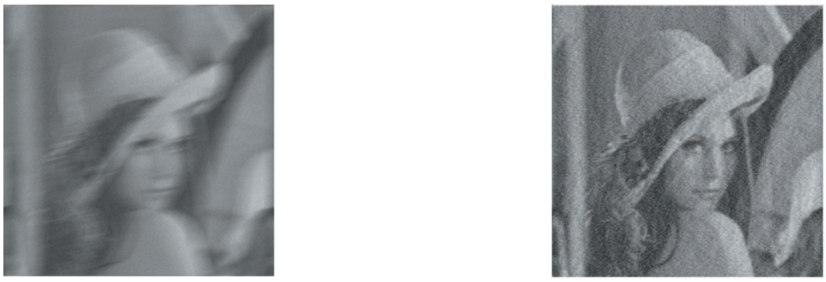
(ii) Result of inverse filtering

Figure 2.13: The effect that additive noise has on inverse filtering

Where $H(u,v)$ is the frequency response of the blur PSF and $S_{gg}(u,v)$, $S_{\eta\eta}(u,v)$ are respectively the mean power spectral density of the degraded image $g(x,y)$ and the noise $\eta(x,y)$. A better interpretation of the Wiener filter can be obtained by writing (2.4) as

$$W(u,v) = \frac{1}{H(u,v)}\left[\frac{\|H(u,v)\|^2}{\|H(u,v)\|^2 + \frac{S_{\eta\eta}(u,v)}{S_{gg}(u,v)}}\right] \tag{2.5}$$

It is easy to see that in the case of very high signal to noise ratio ($SNR = \frac{S_{\eta\eta}(u,v)}{S_{gg}(u,v)}$) the term inside the square brackets approaches 1 and the filter ends up being the inverse filter $\frac{1}{H(u,v)}$. As the noise at certain frequencies increases the term inside the square brackets also drops. This shows how Wiener filtering attenuates frequencies depending on their noise content. The result of applying Wiener filtering to a noisy and blurry image is shown in Figure 2.14. We can see that the result of this filtering produces an image that is much clearer than the original and has not been affected by the noise.



(i) Blurred image with additive noise

(ii) Deblurred image using wiener filtering

Figure 2.14: The result of applying Wiener filtering to an image with motion blur and additive noise

Another approach for non-blind image deconvolution considers the observed image pixels $g[x,y]$ as realizations of random variables that can be described by a joint probability density $P(g, f)$. It then tries to find an image f that maximizes the likelihood $P(g|f)$. Unfortunately this is in general an ill-posed problem. A solution to this problem is to introduce additional constraints on f. Some methods do this by introducing a prior distribution for the clean image $P(f)$, which incorporates any knowledge we have about f. Due to this prior these methods are called *Bayesian deconvolution methods*[11]. The posterior distribution can then be computed using Bayes rule as follows.

$$P(f|g) \propto P(g|f)P(f) \tag{2.6}$$

In order to find the most likely image f from this distribution we have to calculate its mean. However, this is a multi-dimensional distribution whose integral is hard to compute analytically. Therefore sampling methods such as the Gibbs sampler or Metropolis-Hastings are used to estimate the mean.

Finally there is also the regularization approach for doing non-blind image deconvolution. This approach tries to solve the following regularized minimization problem (penalizing large values of **f**), where **f**, **g** are the clean and observed images written in vector form and K is a blurring matrix.

$$\min_{\mathbf{f}} ||K\mathbf{f} - \mathbf{g}||^2 + \alpha\, ||\mathbf{f}||^2, \tag{2.7}$$

It therefore tries to find the original image **f** that after the blur is closest in the least square sense to the obseved image **g**.

2.3.2 Blind Deconvolution Methods

Wiener filtering requires a significant amount of knowledge of the system. It assumes that the power of the noise is known but more importantly it requires exact knowledge of the blurring PSF. In most real applications, however this information is unavailable or is hard to obtain. This is the reason why blind deconvolution techniques have been widely researched. Some of the most representative methods can be found in [12], [13] ,[14].

There are two ways to perform blind image deconvolution. One way is to extract the blur PSF based on exterior information and then proceed to perform non-blind deconvolution using the aforementioned techniques. In this case a parametric blur model may be used to identify the most likely PSF from the observation. The other approach tries to simultaneously estimate the PSF and original image. Most algorithms that do this use an alternating approach to iteratively identify the PSF and the image.

Blind image deconvolution is an extremely ill-posed problem because it requires to solve the following equation

$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y),$$

which has as unknowns the blurring PSF h and the clean image f. *Multiple image blind deconvolution* methods try deal with this problem by using m independent observations of the scene. The problem is therefore described by the following system of equations.

$$\begin{aligned} g_1 &= h_1 \star f + \eta_1(x,y) \\ g_2 &= h_2 \star f + \eta_2(x,y) \\ &\vdots \\ g_m &= h_m \star f + \eta_m(x,y) \end{aligned} \tag{2.8}$$

By doing this these methods now only have to solve a system of m equations with m+1 unknowns, which is more well-posed. A variation of the multiframe approach proposed in [1] will be used in this project and is analysed in Section 3.4.

Chapter 3

Methodology

In this chapter we describe a method to perform video restoration on the underwater surveillance problem described in Chapter 1. This method can be split into the two steps of denoising and deblurring. The details of the algorithm are described and the motivation behind the key design decisions is explained. An effort has been made to use as much information from past frames as possible without making the method insensitive to gradual changes of the scene.

3.1 General Algorithm Description

Our temporal restoration method relies on the fact that the scene in sequential frames does not change too much. In our underwater recordings, frames sometimes have a lot of activity due to fish that are swimming in the sea. The first aim in the frame restoration is to extract as much information about the scene as possible, which is difficult to do when the fish are present. This is the reason why a background subtraction algorithm is used as a preprocessing step to remove fish from the scene. This will aid in both the denoising and deblurring of the scene.

The dirt on the camera lens seems to create non stationary local blurs in the frames. In addition some noise is present from the camera quantization or floating particles. This problem could be treated as a deblurring problem but as demonstrated in Section 2.3 the presence of noise affects the quality of the deconvolution algorithms. Therefore first a denoising step is added in order to improve the quality of the deblurring. This is why emphasis is put on removing the noise without introducing too much additional blur. The algorithm chosen for the noise removal is a patch-based method which uses patches taken from multiple frames. As seen in Figure 3.1 this denoising algorithm will have as inputs the frames produced after applying the background subtraction algorithm on the recordings.

It was mentioned in Chapter 1 that the blur in the sequence can't be considered stationary as it is affected by the water currents and any motion in the scene. Nonetheless, the blur in sequential frames can be considered similar. The PSFs that are responsible for the blurring of each frame in the recordings are unknown so a blind deconvolution technique must be used. The multiframe blind deconvolution method from [1] was chosen for the deblurring. This deblurring was applied to a dictionary containing recent frames produced by the denoising stage mentioned above and produces an estimate of the original scene as well as estimates of the blur PSFs for each frame. It is possible to incorporate information from past clean frames by adding them to the dictionary as shown in Figure 3.1.

Finally, once the PSF estimates of the blur are available they are used to restore the foreground via non-blind deconvolution. The deblurred foreground is then replaced onto the original image estimate. The entire procedure of the method can be seen in Figure 3.1.

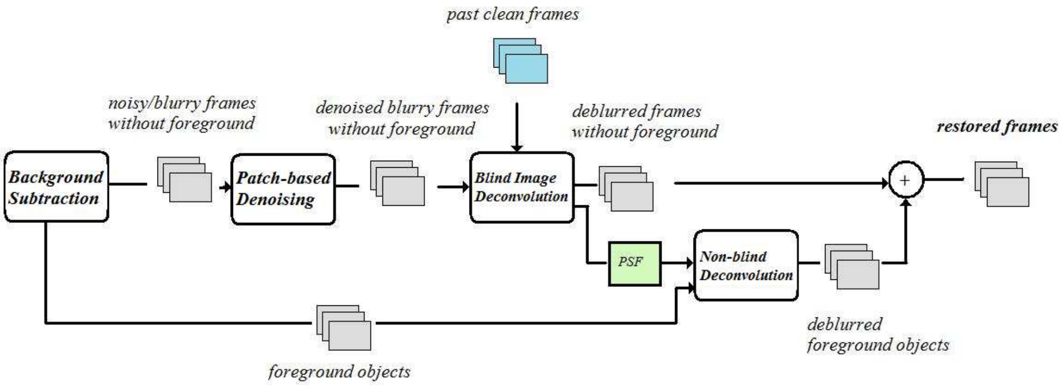

past clean frames
noisy/blurry frames without foreground
denoised blurry frames without foreground
deblurred frames without foreground
restored frames
Background Subtraction
Patch-based Denoising
Blind Image Deconvolution
+
PSF
Non-blind Deconvolution
deblurred foreground objects
foreground objects

Figure 3.1: Steps followed by the proposed method to obtain the reconstructed frames of the video

3.2 Background Subtraction

The fish that are swimming in the frames generally obstruct areas of the scene and are very mobile so they can't be considered part of the background. In order for the method that is proposed in this chapter to work, the fish have to be removed so that they do not interfere with the process identifying the blur PSF. Once the PSFs have been estimated the fish can also be deblurred using non-blind deconvolution, assuming that they are affected by approximately the same blur as the scene.

There are various background subtraction techniques that can be used to obtain the foreground in cases where a static camera is observing a scene. The background subtraction algorithm that is used here for achieving this is the one proposed in [15]. This method builds a statistical model for the background scene and classifies the pixels that do not fit the model as foreground. The classification can be done based on the following threshold R.

$$R = \frac{p(BG|x^{(t)})}{p(FG|x^{(t)})} = \frac{p(x^{(t)}|BG)p(BG)}{p(x^{(t)}|FG)p(FG)} \tag{3.1}$$

The prior probabilities of the foreground and background ($p(FG)$ and $p(BG)$) are set accordingly depending on the knowledge we have of the recorded scene. The density $p(x^{(t)}|FG)$ can be set to uniform if nothing else is known about it. Whether a pixel corresponds to background or foreground is determined by the following decision rule

$$p(x^{(t)}|BG) > c_{thr}, \tag{3.2}$$

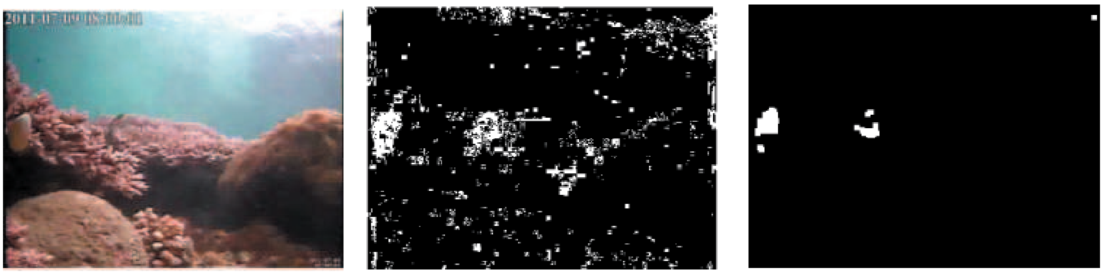
where c_{thr} is a threshold that can be tuned.

The background model is trained on a set of past images $\mathcal{X}$. This estimated model $P(x^{(t)}|\mathcal{X}^{(t)}, BG)$ is assumed to be a mixture of Gaussians as shown below

$$P(x^{(t)}|\mathcal{X}^{(t)}, BG) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x, \mu_m, \sigma_m I) \tag{3.3}$$

Modelling with a multi-modal distribution allows us to account for cases where the background is not stationary but is characterized by small jitter. This is ideal for the scene of our problem, which is affected by underwater currents. This method is also adaptive and is able to cope with changes in the scene illumination and the introduction or removal of background objects. To do this the training set is updated by the addition and removal of images. A suitable time period T is chosen and a training set is formed $\mathcal{X}^{(t)} = \{x^{(t)}, x^{(t-1)} \dots x^{(t-T)}\}$. At every time instant the training set $\mathcal{X}^{(t)}$ is updated and the conditional probability $P(x^{(t)}|\mathcal{X}^{(t)}, BG)$ is recalculated. The parameters for the number of components as well as the number of frames kept in the training set were decided by performing a grid search and judging based on a visual assessment of the masks. The number of gaussians used was 5 and the training set contained 100 frames at each time instant.

This method produces a mask of the foreground which contains speckles due to some pixels being falsely classified as foreground. The mask is then cleaned by performing the open binary operation (erosion followed by dilation of pixels) twice. Sometimes the fish are not fully detected by the algorithm so once the image is cleaned from the speckles the binary operation close (erosion followed by dilation of pixels) is performed three consecutive times to ensure full detection of the fish. This means that the mask sometimes also contains part of the background near the fish. However, this is still preferable to having fish present in the image, which would deteriorate the performance of the denoising and deblurring. The results of the background subtraction can be seen in Figure 3.2. The algorithm has correctly identified the fish on the left of the image, although it has also detected some of the background as well. It must be noted that the background subtraction improves its model with time so after some frames it will very accurately detect the fish (this is shown in the case of additive noise below). The background subtraction was also tested on the blurred frames with slightly less accurate results. The results for a blurred frame can be seen in Figure 3.3

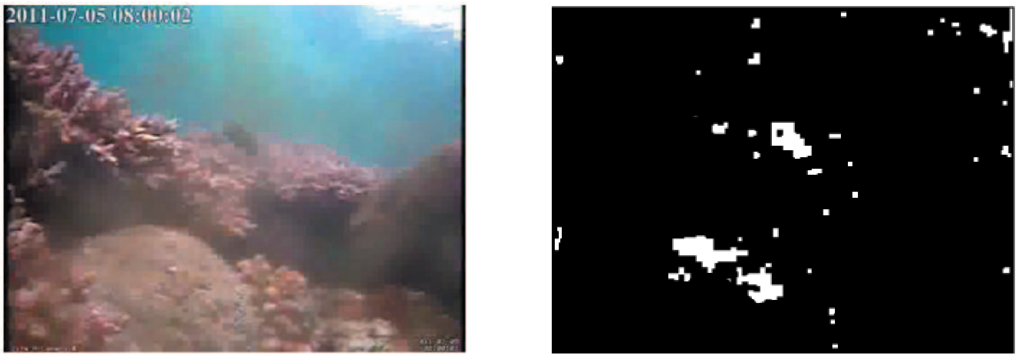


(i) Clean frame (ii) Mask with speckles (iii) Cleaned mask

Figure 3.2: Resulting mask obtained by the background subtraction algorithm in both the clean and dirty version.

The background subtraction also works when noise is presence although it requires a larger transition period until the mask is accurate. Masks from two different frames corrupted by artificial white gaussian noise with 20 units of variance are shown in Figure 3.4.

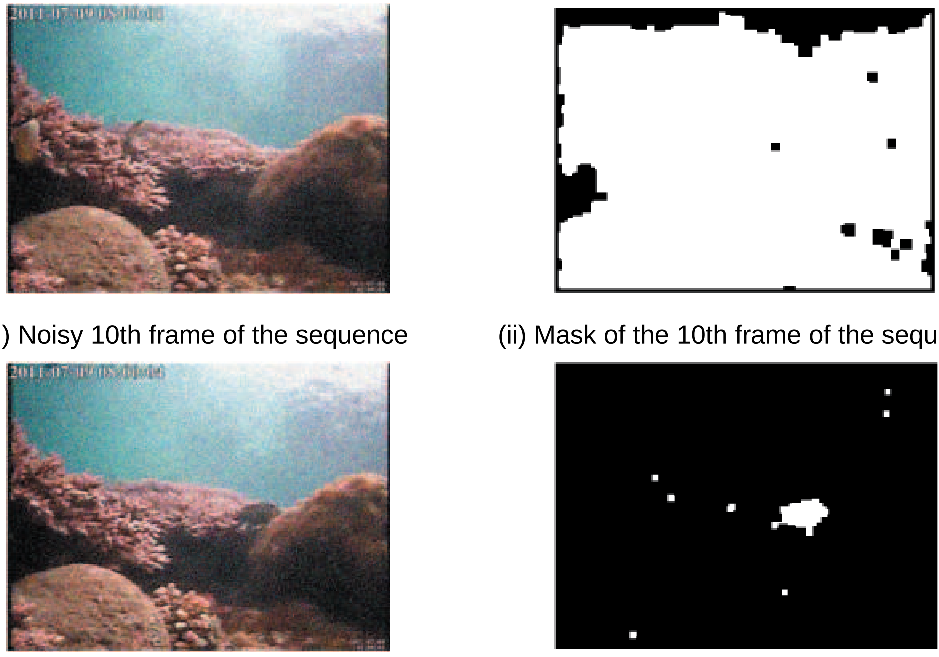
As we can see although the foreground is not correctly detected at the start of the sequence it improves and by the 100th frame it is very accurate. This transition period is not long considering that it corresponds to 4 seconds in a video which runs for hours.



(i) Clean frame

(ii) Mask of foreground

Figure 3.3: Clean mask obtained from using the algorithm on a blurry frame.



(i) Noisy 10th frame of the sequence

(ii) Mask of the 10th frame of the sequence

(iii) Noisy 100th frame in the sequence

(iv) Mask of the 100th frame of the sequence

Figure 3.4: Improvement of the model of the background subtraction algorithm with time. Masks of noisy frames improve with time as the model gets trained on more and more data.

3.3 Denoising Step

The goal for the denoising stage is to create a method that uses the past frames of the sequence. It is reasoable to assume that noise will not affect in the same pixels over time, which is why we expect a temporal method to be better as far as denoising is concerned. A patch-based algorithm is proposed due to the simplicity with which it can be modified to deal with videos. The stages of this step are explained in the following sections.

3.3.1 Grouping Stage

Typical variations of the BM3D algorithm used for video denoising usually substitute the spacial search for similar patches with a temporal-spacial search [16]. This search can be very computationally expensive and is usually narrowed down to a smaller area to improve the speed of the algorithm. In the surveillance problem however it is fairly certain that the scene being monitored does not change very much with time. It is safe to assume that patches located in the same place in different frames will be similar. We therefore propose a method that does not search for similar frames but instead groups together patches that are located in the same area at different frames. In order to produce a method that is adaptive and can cope with gradual natural changes in the scene such as lighting changes or slight swaying of the scenery blocks were created using a dictionary containing only a few of the latest frames.

The aforementioned similarity assumption is invalid if there are fish present in the patches. The background subtraction is responsible for finding any fish and removing them. Initially the idea of not including patches with fish in the groups was tested but this sometimes resulted in having groups with no patches due to the constant presence of fish. This meant that the group-based filtering couldn't be applied. It was therefore decided to keep all the patches but replace the foreground pixels in them with the median of the pixel values of another dictionary that is initially larger than the one used for the grouping.

In general, this approach greatly decreases the complexity of the algorithm by avoiding the computationally demanding search step and provides us with almost identical patches within a block. An example of patches that are grouped together using this method can be seen in Figure 3.5.

It is evident from the figure that the blocks grouped by this method look very similar. The blocks that were generated did not contain overlapping patches for simplicity. The classical BM3D algorithm uses overlapping patches to generate more similar patches, however in the case of the surveillance problem this is rarely an issue since there are many similar patches.

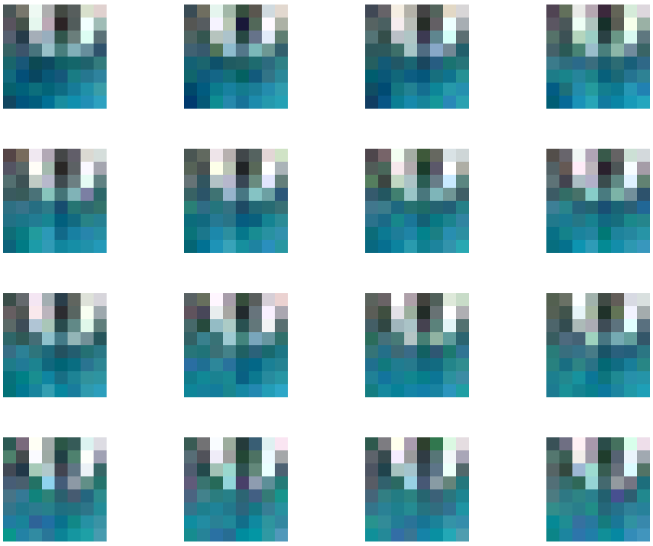


Figure 3.5: Patches found within the same block using the grouping algorithm.

3.3.2 The Collaborative Filtering Stage

The second stage in the denoising step is that of collaborative filtering and is basically the stage where the noise reduction takes place. Once the blocks are obtained a 3D linear transform is applied to each block in order to get the frequency representation of the image. There are multiple linear transforms that can be used but it is convenient to use a separable transform. Separable means that the N dimensional transform can be calculated as a separable product of N sequential 1D transforms along each dimension. The transform that was chosen was the Discrete Cosine Transform (DCT) who's 2D version is shown below

$$X_{k_1,k_2} = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} x_{i,j} \cos\left[\frac{\pi}{N_1}\left(i+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(j+\frac{1}{2}\right)k_2\right]. \tag{3.4}$$

It is evident from (3.4) that the DCT is separable, which allows for the calculation of the 3D DCT as one 2D DCT for each patch in the block and one 1D DCT along the "temporal" dimension of the block.

Once the transform coefficients have been found they are compared to a threshold λ_{thr} and if they are smaller they are set to zero. This is the same hard-thresholding used in the classical BM3D algorithm, which helps attenuate the noise. The parameter λ_{thr} depends on how much smoothing we wish to apply. Large values of λ_{thr} will mean better denoising but more blurring of the edges and small values of λ_{thr} will result in persistent noise. The optimal value for λ_{thr} for most cases was found to be 0.12.

Finally the denoised blocks are obtained through the inverse linear transform. The filtering procedure described above is used in the case of gray scale images. In the case of color images the collaborative filtering must be applied to each color channel separately. The choice of colorspace for the images does not affect the algorithm so the standard RGB was used.

3.3.3 Aggregation Stage

The blocks obtained after the collaborative filtering contain the estimates of their patches. The easiest way of reconstructing the denoised image would be to concatenate the patch estimates that correspond to the current frame. The result of this operation can be viewed in Figure 3.6.

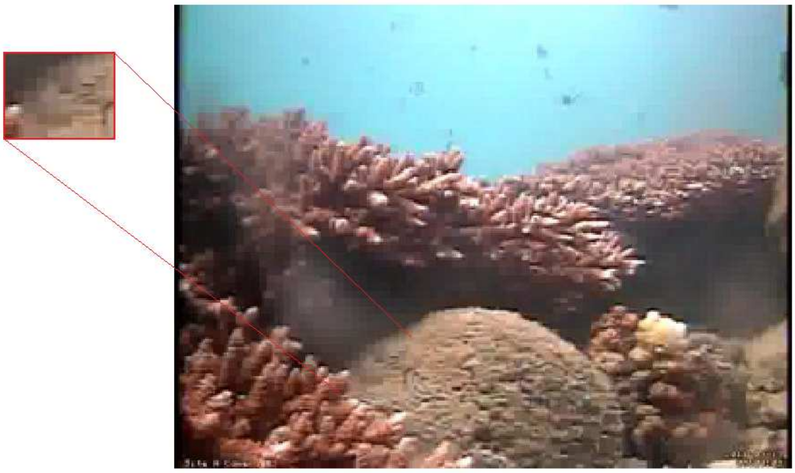


Figure 3.6: Reconstruction of a frame using only the patch estimates from the block that correspond to that frame. The resulting image suffers from artifacts.

It is immediately noticeable that the noise has been successfully dealt with. The reconstruction, however seems to have slightly unnatural transitions from patch to patch. This happens because blocks will have a different number of retained coefficients depending on the homogeneity and the amount of noise that is present in them. Some blocks will therefore lose more detail than others. This effect is usual for patch based methods using magnitude thresholding. For this reason aggregation of the patches similar to the one in BM3D is performed. One can consider this aggregation as a form of weighted averaging of pixels that are in the same position but in different patches of a block. In the case of non-overlapping blocks then weights will be equal and this will be the same as applying smoothing via a mean filter for each pixel across multiple frames. The effects of this averaging are shown in Figure 3.7.

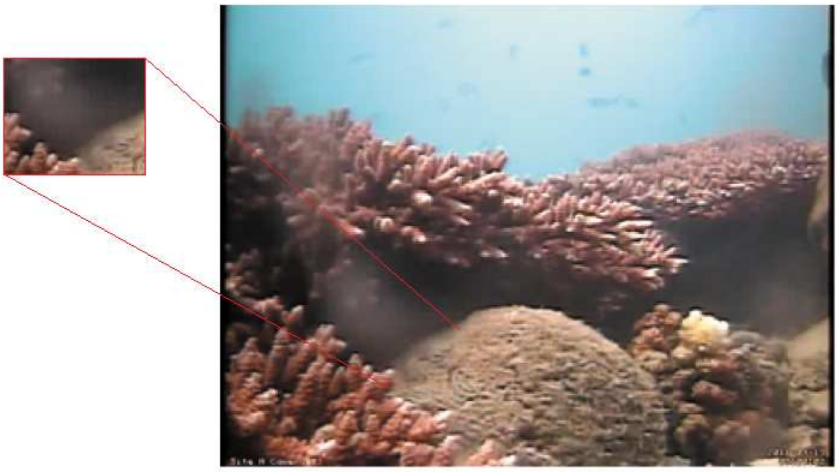


Figure 3.7: Reconstruction of a frame using aggregation of all the patch estimates of the blocks.

3.4 Deblurring Step

Once the frames have been denoised then the deblurring step can take place. For this step we use the algorithm proposed in [17] and [1]. This algorithm is a multi-frame blind deconvolution method that makes use of multiple images of the same scene taken from slightly different angles (slightly misaligned images). This method was chosen for two reasons. Firstly, it can be used with consecutive frames from our recordings as well as clean frames. Using clean frames allows for better estimation of the image as well as the blur. Secondly, it accounts for any swaying motion and minor changes in the scene caused by underwater currents in the scene.

3.4.1 Method Description

It was mentioned previously that this deblurring method uses a dictionary of recent frames from the video in order to deblur the current frame. It is therefore required to solve the following ill-posed system of equations for all PSFs $h_1, h_2 \cdots h_m$ and the original image f.

$$\begin{aligned} g_1 &= h_1 \star f + \eta_1(x,y) \\ g_2 &= h_2 \star f + \eta_2(x,y) \\ &\vdots \\ g_m &= h_m \star f + \eta_m(x,y) \end{aligned} \tag{3.5}$$

In order to solve this system the method imposes certain constraints to the problem by minimizing a regularized energy function shown below.

$$E(f,h_1,\cdots h_m) = \frac{1}{2}\sum_{i=1}^{m} \|h_i \star f - g_i\|^2 + \lambda Q(f) + \gamma R(h_1,\cdots h_m), \tag{3.6}$$

where $Q(f)$ and $R(h_1,\cdots h_m)$ are regularization terms that impose constraints on the original image and PSFs respectively. The parameters λ and γ are positive numbers which penalize the solutions of f and h_i. In order to find the a minimizer of (3.6) alternate minimizations of E are performed with respect to f and $h_1,\cdots h_m$. The terms $Q(f)$ and $R(h_1,\cdots h_m)$ are chosen to be quadratic and therefore convex, which makes finding the derivatives $\underset{f}{\nabla} E$ and $\underset{h}{\nabla} E$ easier. This helps the alternating minimization algorithm (AM) perform a variation of steepest descent in order to minimize the energy function.

AM first descends in the image subspace until it reaches a minimum($\underset{f}{\nabla} E = 0$) and then performs optimization in the blur subspace in a direction that is orthogonal to the one it had previously. This is repeated until convergence. The iterative steps are summarized by the equations shown below

- 1st step: $f^n = \arg\min_{f} E(f^{n-1}, \{h_1,\cdots h_m\}^{n-1})$
- 2nd step: $\{h_1,\cdots h_m\}^n = \arg\min_{\{h_1,\cdots h_m\}} E(f^n, \{h_1,\cdots h_m\}^{n-1})$

These two equations end up being linear and are easy to solve. The energy function is not a convex function with respect to f and h_i so it is not guaranteed to converge to a global optimum. In the first step of AM, the minimization is done using the conjugate gradients method and then the solutions are filtered by the constraints. The second step involves calculating the blur PSF, which is much smaller in size than the image and therefore can be performed as a constrained minimization without being too computationally demanding.

The values of the regularization parameters λ and γ can be determined through analysis but in our case are tuned manually to suit the problem at hand via visual assessment of the output frames. The optimal values of these parameters vary from problem to problem and have to be tuned for each sequence.

It must be noted that this method assumes that the size of the PSFs is known. However the method will still function well if the size is overestimated (the extra coefficients will be very close to zero), although there is the fear of over-fitting. However if the size is underestimated then the method will not be able to deal with the problem well. Fortunately the size of the PSFs can also be tuned via visual assessment.

Once the blur PSF has been estimated it can be used to deblur the foreground that was removed by the background subtraction. This assumes that the fish will have been blurred in the same way as the rest of the image. The deblurred foreground is then overlaid on top of the deblurred frame to produce the final estimates.

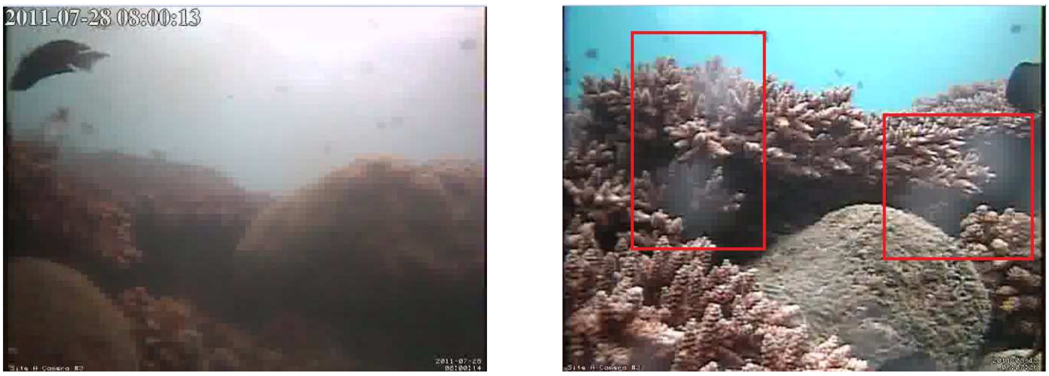
3.4.2 Method Variations

The above method can be implemented as described in the previous section to deblur the frames, however some different variations of the method were also considered.

3.4.2.1 Patch Based Variation

The first variation of the method tries to adopt a patch based approach by dividing the frames into patches and forming blocks using the corresponding patches from other frames in the dictionary. It then proceeds to apply the deconvolution algorithm on each block separately to produce an estimate of the block as well as an estimate of the local PSF. The reason behind this decision was to deal with the local blur that is present in some sequences. Due to the fact that both local and global blur were present in our dataset (Figure 3.8) the developed algorithm gives the user the choice between patch-based or whole-image deconvolution.

The subject of local blur is a very difficult one, which has not yet been researched. The problem with local blur is that it is most likely caused by multiple PSFs that operate on small patches of the image. If the blind image deconvolution algorithm is applied to the whole frame then it will try to attribute the blur that is present in the image to only one PSF. This is futile since the same PSF can't be responsible for a blurry part of an image as well as a non blurry part. This is the reason why it is worth trying to use a patch based approach.



(i) Frame with global blur

(ii) Frame containing local blur

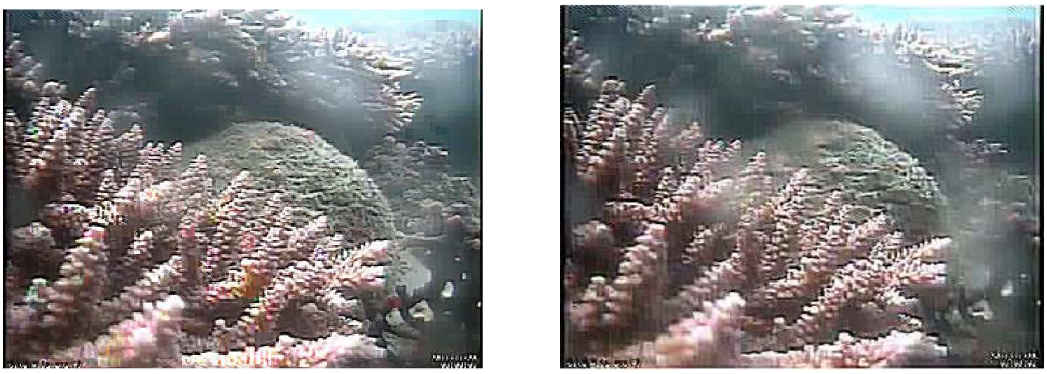
Figure 3.8: Examples of local and global blur in the recordings.

3.4.2.2 Variation using Clean Frames

The second variation of the algorithm is based on the fact that knowledge of when the camera was last cleaned is available. Therefore, clean frames can also be inserted in the deblurring algorithm's dictionary so as to extract more information about the scene. By adding a clean frame we hope to influence the algorithm in producing a reconstruction that is closer to the it. This can only be done if the algorithm correctly understands that the clean frame is indeed close to the original scene and thus assigns to it a function that is close to the Dirac Delta, since convolution of any signal with the Dirac Delta produces the same signal.

If the method described in section 3.4.1 is used as it is, it will produce estimates of the PSFs and the original scene that are directly influenced by the clean frame. Thus, estimate of the original scene will end up being an average of the deblurred frames and the clean frame. It would be preferred ,however, if our estimate of the original scene is not directly influenced by clean frames. The reason is that clean frames are taken from too far in past and do not reflect the current situation of the seabed. This would make our algorithm insensitive to any minor changes in the scenery such as slight movements and illumination changes.

A more indirect way for using the information of the clean frames must be found. It is proposed that only the PSF estimates, obtained by the multi-frame deconvolution algorithm, be used and not the estimate of the scene itself (since it contains part of the clean image). The PSF estimate of the blur in the current frame can then be used in conjunction with the current blurred frame to produce an estimate of the original scene via non-blind deconvolution. This way the clean frame affects our estimate of the original scene indirectly through the blurring function. Any changes in the illumination or any details that were present in the blurry frames will therefore be maintained. The reconstruction that is obtained from directly using the clean frames in the estimate and the reconstruction that indirectly uses the clean frames can be seen in Figure 3.9.



(i) Reconstruction obtained from directly using the clean frames in the estimate.

(ii) Reconstruction obtained from indirectly using the clean frames in the estimate.

Figure 3.9: Examples of reconstructions obtained from i)directly ii) indirectly using clean frames.

The reconstruction that is obtained directly from the clean frames seems better than the one that indirectly uses them. However, this result is artificial and is caused by the averaging of the clean and blurry frames(this is why some of the local blur also disappears). In order to produce an adaptive algorithm that correctly represents the current state of the scene the indirect approach must be used, which does not produce much inferior results.

Chapter 4

Experiments and Evaluation

In this chapter the experiments held with the proposed method will be presented and assessed. Experiments were held to show the performance of the proposed method on the blurry sequences. Additional tests with artificial deterioration were also held to determine the performance of each stage in our method (denoising/deblurring). The outline of the chapter is as follows. First the denoising algorithm will be tested on sets with generated noise and compared to other methods. Then the performance of the deblurring algorithm is assessed for frames that were corrupted by artificial local and global blur. Finally the results produced by the whole algorithm on our real data will be presented and discussed.

4.1 Methods of Evaluation

It is very important to establish a measure with which to evaluate the quality of the video reconstruction. When assessing the quality of frames one must usually have an idea of what the original noiseless frames look like. The more similar the reconstructed video is to what is assumed to be the original sequence the better the quality it has. Two videos are considered similar if their respective frames are identical on average.

If we possess the original clean frames then we can use measures such as the Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) to assess the quality if the reconstruction. These measures are termed “objective” due to the fact that they do not depend on human judgement, which varies depending on a persons personal aesthetics, but are based on a pixel by pixel differences of the frames.

The mean squared error (MSE) has been the most widely used quantitative performance metric in signal processing. It is computed according to formula (4.1).

$$MSE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(S_{original}(i,j) - S_{reconstructed}(i,j))^2 \tag{4.1}$$

PSNR is another form of the MSE whose use is even more widespread for image applications. The formula for this is seen in (4.2).

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{4.2}$$

where MAX_I is the maximum pixel value in the image. The MSE is widely used in signal processing due to its simplicity, the fact that it is parameter free and inexpensive to compute. It is also a natural way to describe signals because it represents the energy of the signal. PSNR and MSE are different forms of the same thing although the reader must be careful to notice that while a low MSE signifies a better similarity the opposite holds for PSNR (a high PSNR is better).

It must be mentioned that there are many cases where MSE/PSNR are wrong in assessing the quality of an image. This stems from the fact that they are based on pixel by pixel differences. Thus the are affected by several factors such as illumination changes, translation etc. This can be understood via an example from [18] shown in Figure 4.1.

This example shows why relying only on “objective” metrics might not be the correct thing to do when judging the quality of the reconstruction. For this reason examples of frames will be presented to back up any claims about quality. Surveys where also conducted with people who judged the quality of the image. Of course when the data permits the PSNR will also be presented.

4.2 Denoising Experiments

Our denoising algorithm was tested on several sequences with varying amounts of blur that where corrupted by additive noise in order to determine how well it performs against other popular denoising techniques. Due to the fact that the noise levels in the initial video were not very high the denoising experiments were held with frames that where artificially corrupted by noise. This also allows for comparison of the results of the denoising with the original frames. Experiments were held using various types of noise and noise levels. The proposed denoising algorithm is compared to the median filter (using a 3-by-3 neighbourhood) and the BM3D method. These two denosing methods where chosen for the comparison because of the fact that they are the state of

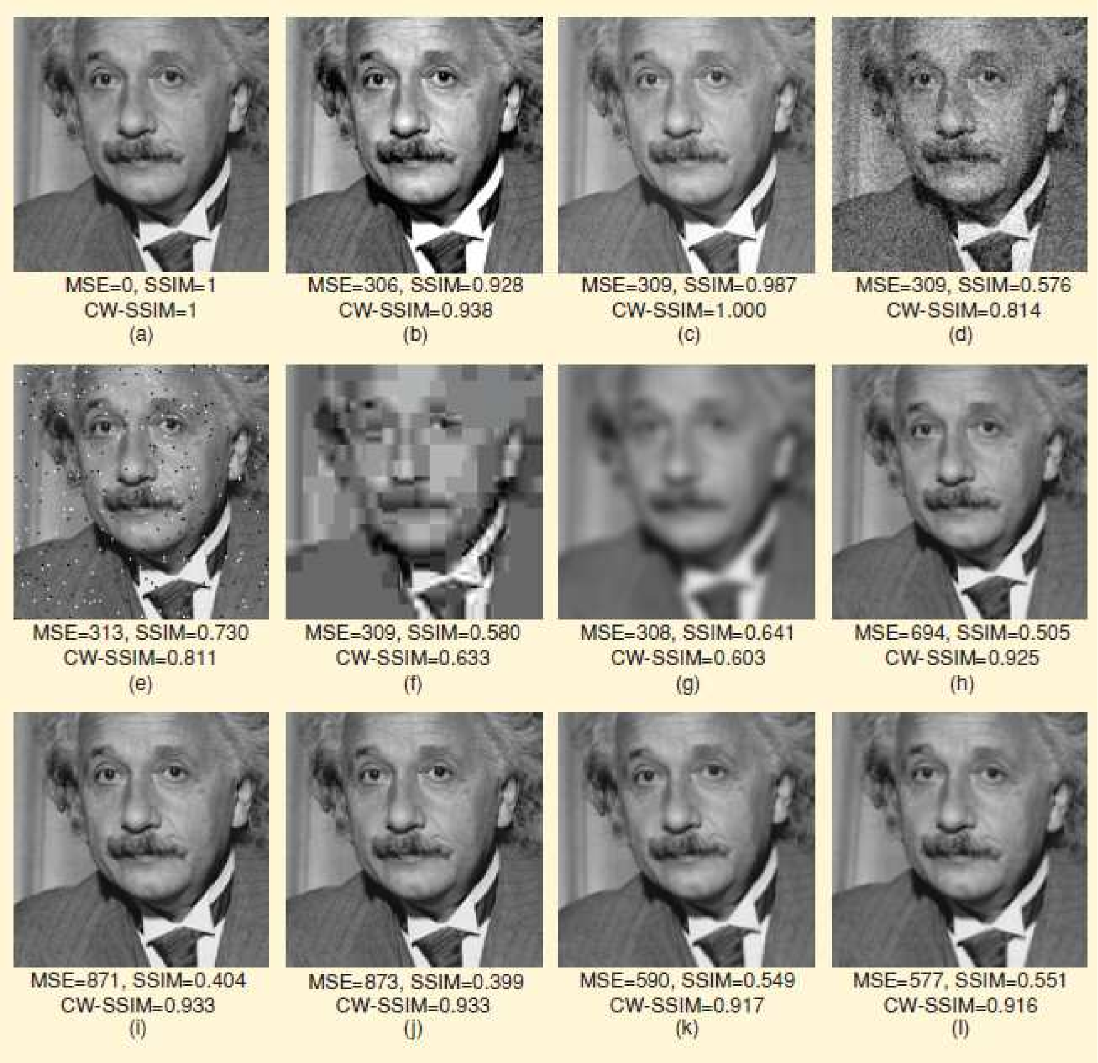


Figure 4.1: Comparison of image fidelity measures for Einstein image altered with different types of distortions. (a) Reference image.(b) Mean contrast stretch. (c) Luminance shift. (d) Gaussian noise contamination. (e) Impulsive noise contamination. (f) JPEG compression. (g) Blurring. (h) Spatial scaling (zooming out). (i) Spatial shift (to the right). (j) Spatial shift (to the left). (k) Rotation (counter-clockwise). (l) Rotation (clockwise) taken from [18]

the art spacial denoising methods for different types of noise. Our proposed denoising algorithm has to make use of the background subtraction to remove the fish from the image. The fish were then replaced for the comparison with the original frames. All the images used in these experiments had pixel intensities stored as double precision floating point numbers in the range of $[0,1]$.

4.2.1 Experiment with Additive Gaussian Noise

4.2.1.1 Frames Corrupted by Weak Noise

In this experiment a sequence of frames was corrupted with additive zero-mean gaussian noise with a standard deviation of 0.04[1] . The chosen test sequence contained 400 frames and was assessed using the Mean Peak Signal-to-Noise Ratio. The results are shown in Table 4.1 and Figure 4.2[2] .

Algorithm	MPSNR (R channel)	MPSNR (G channel)	MPSNR (B channel)
BM3D	35.8792 dB	36.4760 dB	36.4194 dB
Median filter	25.5234 dB	25.5068 dB	25.5633 dB
Proposed method	32.8452 dB	33.5798 dB	33.5922 dB

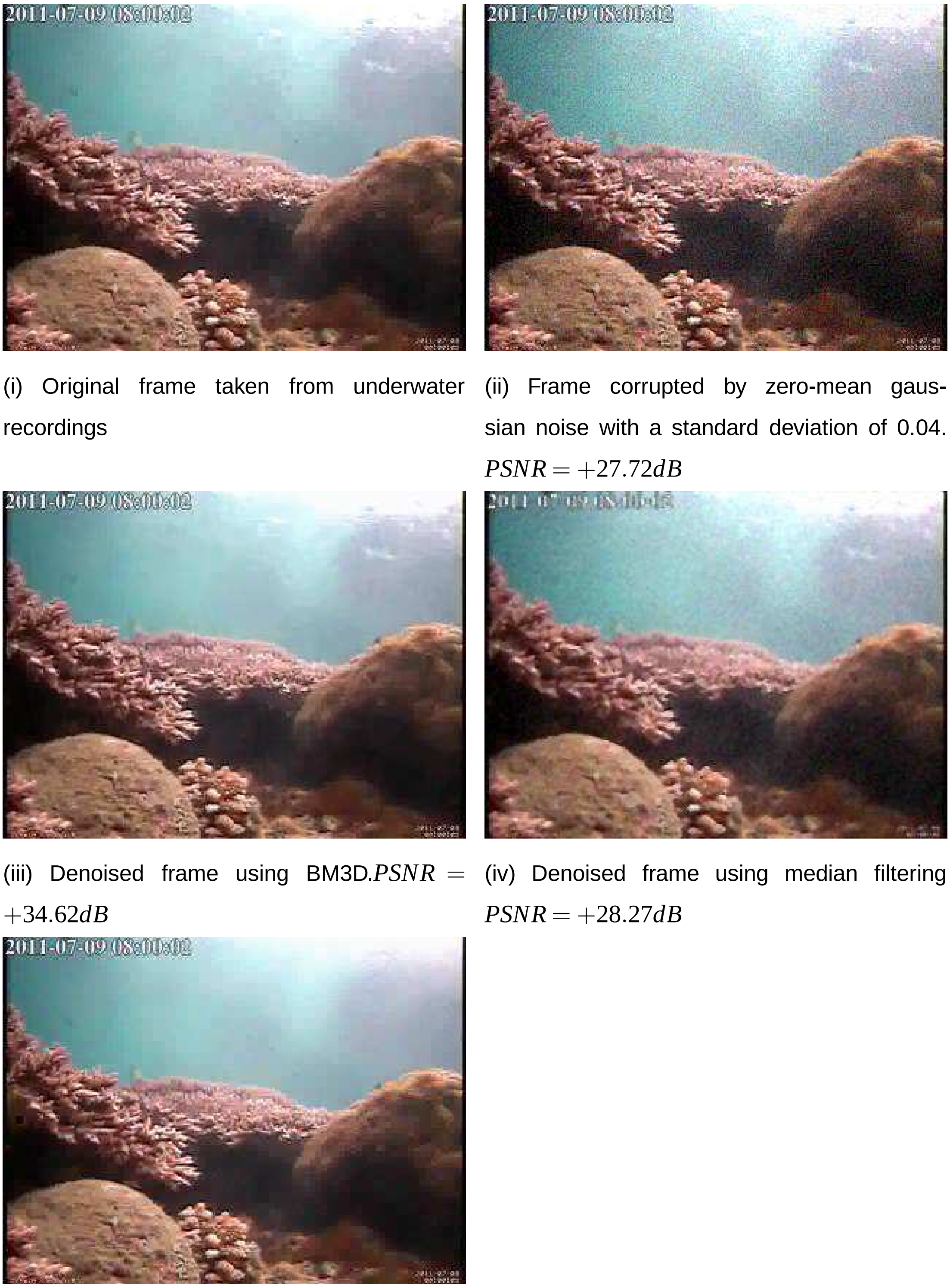
Table 4.1: PSNR of the tested methods for each channel averaged over 400 frames

The results show that the patch based methods clearly outperform the median filter. The proposed variation of BM3D gives a slightly worse PSNR compared to the classical BM3D algorithm. The difference between the PSNR of the two images is small enough so that it does not signify that one is necessarily better than the other. In fact the difference could be attributed partly to the fact that our proposed method is highly dependent on the background subtraction. If the background subtraction fails then the fish that will be present in the patches will affect the denoising and the PSNR. This phenomenon is very rare although there are some frames especially in the beginning of the sequence (where background subtraction is still not very accurate) where this happens. An example of a case where the background subtraction fails is shown in Figure 4.3.

In order to obtain a qualitative measure for the quality of the reconstructions a survey was held where people where shown frames of each reconstruction and where asked to vote for the one that was closest to the original. In total 11 people were shown frames of the original sequence and frames that were produced using each of the methods and to choose the method that produced the closest reconstruction to the original. Frames were chosen so that they did not contain significant errors from the background subtraction. The survey revealed that 55% of those asked prefered the reconstruction of the classicl BM3D, followed by 45% who chose our proposed approach. Median filtering received no votes.

[1] Images have pixel intensities in the range [0,1]

[2] The average PSNR for all three channels is presented for simplicity when they are all similar.



(i) Original frame taken from underwater recordings

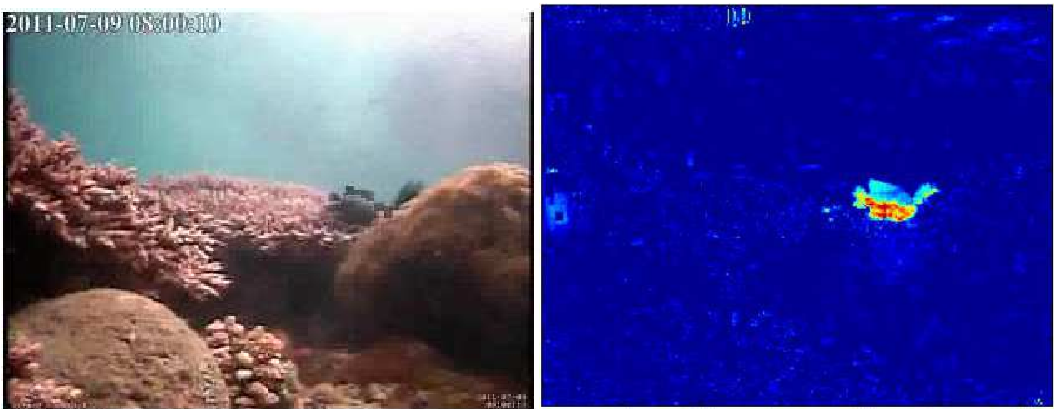
(ii) Frame corrupted by zero-mean gaussian noise with a standard deviation of 0.04. $PSNR = +27.72dB$

(iii) Denoised frame using BM3D.$PSNR = +34.62dB$

(iv) Denoised frame using median filtering $PSNR = +28.27dB$

(v) Denoised frame using the proposed method $PSNR = +34.88dB$

Figure 4.2: Example of denoising of a frame from the sequence for the proposed method, BM3D and median filtering



(i) Frame denoised with bad background subtraction $PSNR = +25.31dB$

(ii) Intensity image of pixel by pixel difference for cases when the background subtraction fails

Figure 4.3: Example showing how a non detected fish causes bad image reconstruction

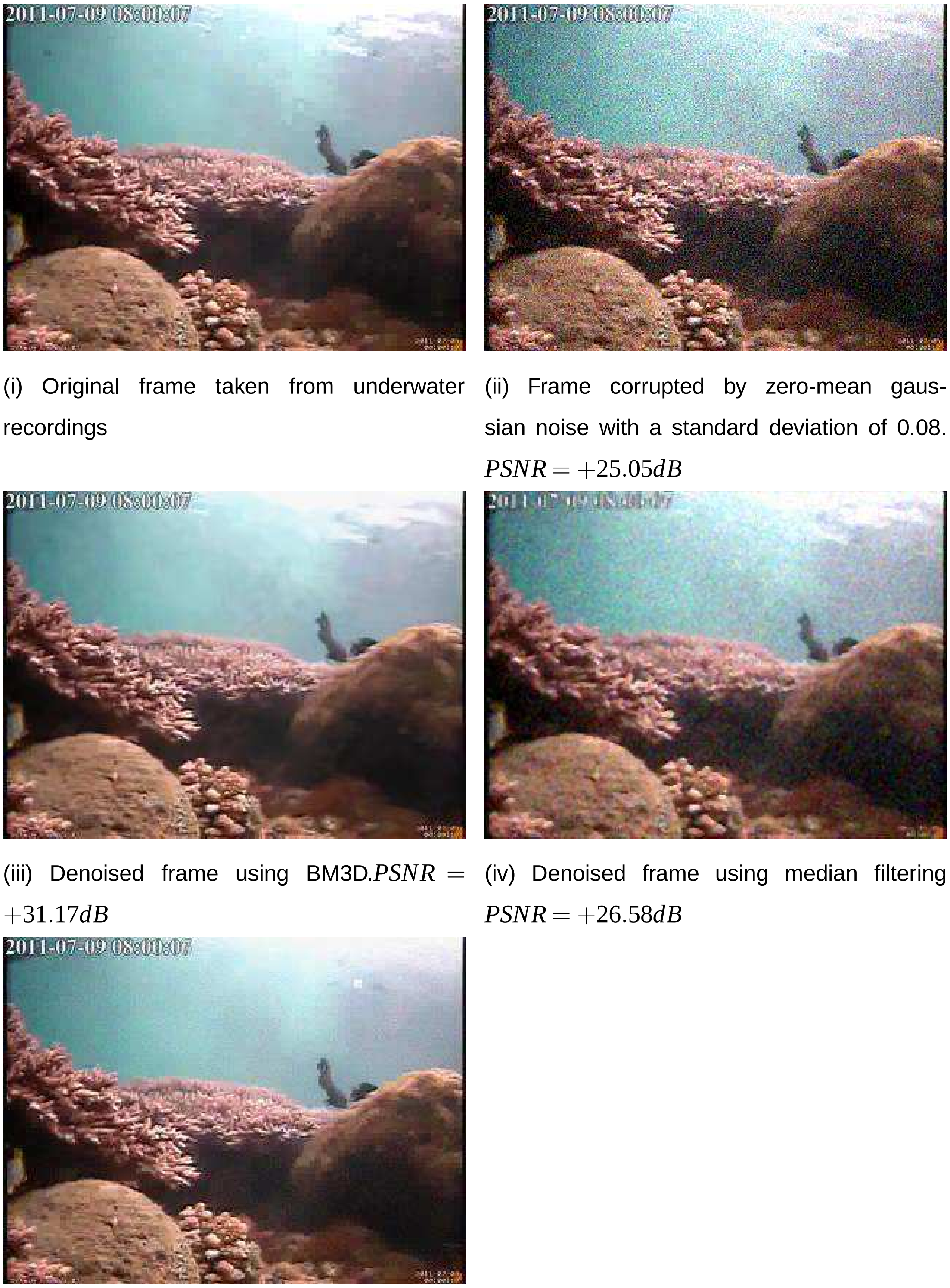
4.2.1.2 Frames Corrupted by Strong Noise

In order to test the denoising capabilities of our algorithm further it was tested on a very noisy sequence and its performance was compared to other denoising methods. Frames were corrupted with an additive zero-mean Gaussian noise with a standard deviation of 0.08. By adding more noise we can also test the robustness of the background subtraction. The PSNR was not measured for the initial frames where the background subtraction was still initialising. The mean PSNR of the last 200 frames in a sequence of 400 is shown in Table 4.2. An example of the noisy and denoised frames from each method can be seen in fig:examples 2nd experiment.

Algorithm	MPSNR (R channel)	MPSNR (G channel)	MPSNR (B channel)
BM3D	31.8202 dB	32.2101 dB	32.1059 dB
Median filter	21.5965 dB	21.6732 dB	21.7424 dB
Proposed method	31.9018 dB	32.3546 dB	32.3765 dB

Table 4.2: PSNR of the tested methods for each channel averaged over 200 frames

Once again the results show that median filtering is far worse than the patch based methods producing a very blurry result that is still affected by noise. Our proposed variation of BM3D performs similarly to the classical BM3D approach. The mean PSNR for our method is slightly better than that of the BM3D approach, although once again this difference is negligible. This is why we have also resorted to a survey



(i) Original frame taken from underwater recordings

(ii) Frame corrupted by zero-mean gaussian noise with a standard deviation of 0.08. $PSNR = +25.05dB$

(iii) Denoised frame using BM3D.$PSNR = +31.17dB$

(iv) Denoised frame using median filtering $PSNR = +26.58dB$

(v) Denoised frame using the proposed method $PSNR = +33.1dB$

Figure 4.4: Example of denoising of a frame from the sequence for the proposed method, BM3D and median filtering ($3 \times 3 neigbourhood$)

where it was revealed that 64% of those asked prefered our proposed approach to the other methods. The classical BM3D received 36% of the votes and median filtering received no votes. It is interesting to note that the mean PSNR our approach was not affected as much as the others by the increase in noise. The experiment with salt and pepper noise that follows will explain why this happens.

4.2.1.3 Results for Various Sequences

In order to get a better understanding of how the denoising performs for various sequences and with different levels of noise a series of experiments were performed. The PSNR for the three methods that were tested are shown in Table 4.3.

Data set	BM3D	Median filter	Proposed method
1	35.1287 dB	24.8604 dB	28.1805 dB
2	30.8360 dB	21.4010 dB	27.7212 dB
3	36.6946 dB	26.4626 dB	30.1201 dB
4	32.7210 dB	21.8886 dB	29.5727 dB
5	38.7728 dB	26.9526 dB	36.0280 dB

Table 4.3: PSNR for the denoising methods tested on various datasets.

The results are similar across all the recordings. The classical BM3D approach usually has a slightly larger PSNR than our proposed method which can partially be explained by the failing of the background subtraction for certain frames. In order to provide a more subjective qualitative assessment a survey similar to the ones performed in the previous sections was held for each data set. The results of the surveys that were held for the qualitative assessment of the frames are shown in Table 4.4.

The surveys reveal that for the majority of the tests our proposed method produced

Data set	BM3D	Median filter	Proposed method
1	28%	0%	72%
2	36%	0%	64%
3	45%	9%	56%
4	36%	0%	64%
5	56%	0%	44%

Table 4.4: Percentages of people who voted for each of the competing methods.

a more aesthetically pleasing result, although the number of people that preferred the the classical BM3D method is not negligible. The main problem that people found with the classical BM3D approach was that it tended to blur the details in the scene a bit more than our approach. This can be attributed to the fact that the classical BM3D groups together patches that may be vary slightly in some details and these details might be lost during the filtering stage.

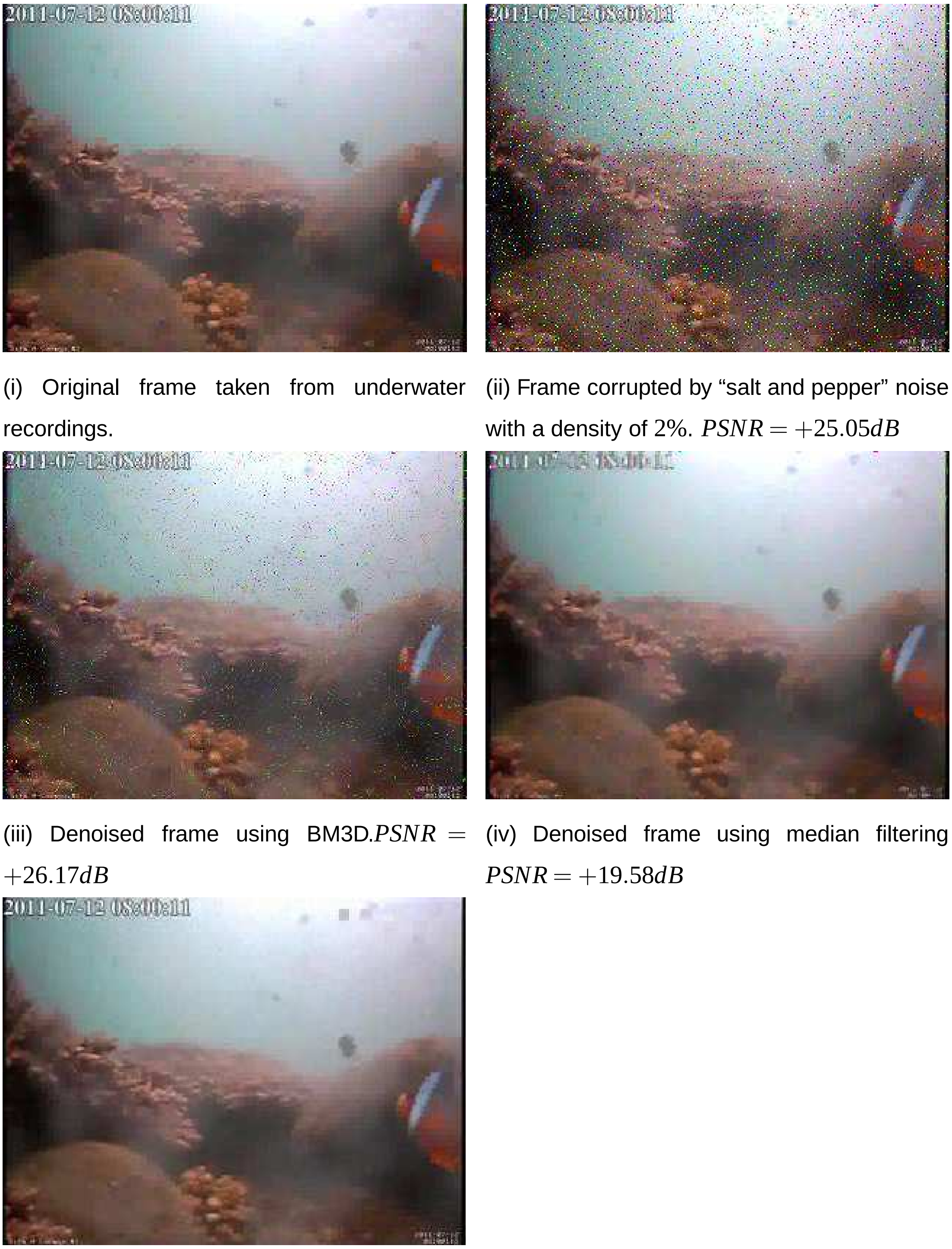
4.2.2 Experiments with Salt and Pepper Noise

This experiment deals with the removal of salt and pepper noise. This noise consists of random light and dark pixels appearing in the images. This type of noise is typically caused by timing errors in the digitization process. The methods were all tested in a sequence of 400 frames that were corrupted by “salt and pepper” type noise which affected 2 percent of the frame pixels. An example of a noisy frame and denoised frames from each method can be seen in Figure 4.5.

Algorithm	MPSNR (R channel)	MPSNR (G channel)	MPSNR (B channel)
BM3D	27.1202 dB	26.2101 dB	26.1059 dB
Median filter	19.5965 dB	18.6732 dB	18.7424 dB
Proposed method	30.9018 dB	30.3546 dB	31.3765 dB

Table 4.5: PSNR of the tested methods for each channel averaged over 200 frames

It is evident from the above examples that the classical BM3D can’t cope with the “salt and pepper” noise. This happens because “salt and pepper” noise consists of outliers that will greatly affect the grouping stage of the classical BM3D. Due to the fact that BM3D fails in its search to find similar patches it either can’t create blocks or creates blocks of patches that are not very similar. The median filtering is known to be one of the best methods for dealing with “salt and pepper” noise. However, as we can see the proposed approach also deals well with the noise. This happens because groups are formed without performing a search. Once the blocks are formed then the noise will be filtered out during the hard-thresholding stage of our algorithm. In addition our approach manages to better maintain the details of the frames compared to median filtering. A survey held with 11 people revealed that our 70% of those asked preferred our proposed approach to that of median filtering, which received 30% of the votes. The classical BM3D method as expected received no votes.



(i) Original frame taken from underwater recordings.

(ii) Frame corrupted by “salt and pepper” noise with a density of 2%. $PSNR = +25.05dB$

(iii) Denoised frame using BM3D.$PSNR = +26.17dB$

(iv) Denoised frame using median filtering $PSNR = +19.58dB$

(v) Denoised frame using the proposed method $PSNR = +32.1dB$

Figure 4.5: Example of denoising of a frame from the sequence for the proposed method, BM3D and median filtering ($3 \times 3 neigbourhood$)

4.3 Deblurring experiments

In order to test our deblurring algorithm we conducted experiments using the denoised non-blurry frames, which contained no fish, obtained from the previous steps. Then various types of artificial blur were added to them. Experiments where conducted with both artificial global and local blur.

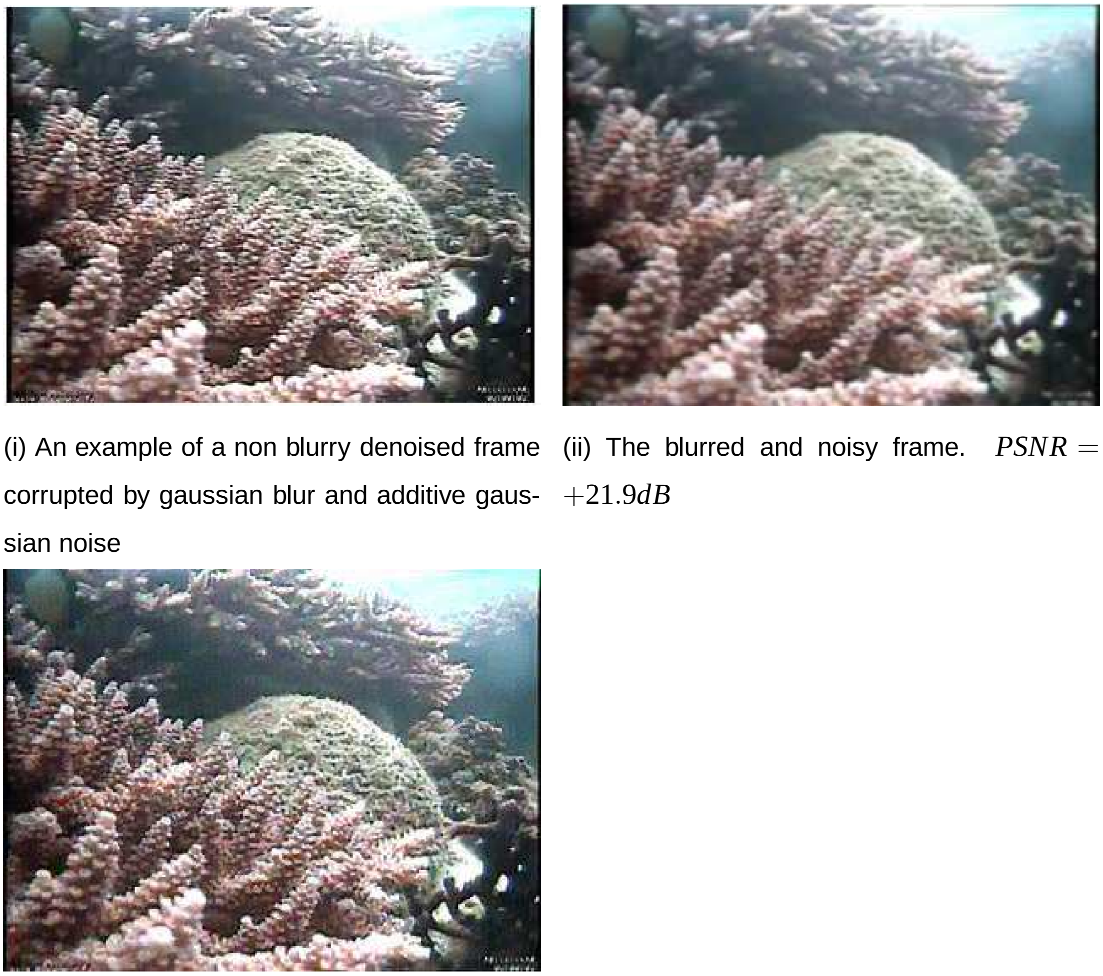
4.3.1 Performance on Artificial Global Blur

This section presents experiments that were performed on frames that were corrupted by adding artificial global blur. In order to cover most cases of blur that might have occurred in our frames, experiments were held for three different types of global blur. The first type of blur is one that has a Gaussian PSF that stays the same throughout all the frames of the video. The second type of blur examines the possibility of a PSF that is close to Gaussian but exhibits some deviation from frame to frame. Finally, the last experiment deals with the case of motion blur, where the direction of the motion changes from frame to frame.

4.3.1.1 Gaussian Blur throughout all Frames

In this experiment we blurred three of our denoised frames containing no foreground and blurred them with the same Gaussian PSF with a variance of 1 and corrupted by zero-mean Gaussian noise with a variance of 0.0001. We call this a global blur because it is not limited to a section of the image but affects the entire image. The effects of this blur as well as the deblurred image we get from the proposed method are shown in Figure 4.6.

The reconstruction seems to be quite sharp and similar to the original and has a PSNR that is larger than that of the corresponding blurry frame. It is expected that the algorithm will have also recovered the PSF of the blur. An example of the recovered PSFs is shown beside the original PSF of the blur in Figure 4.7. From the comparison of these two images we can see that the algorithm has managed to correctly estimate the form of the PSF. The two PSFs were compared by the pixel-by-pixel measures and were found to have a MSE of 0.0024 and PSNR of $+40.14dB$.

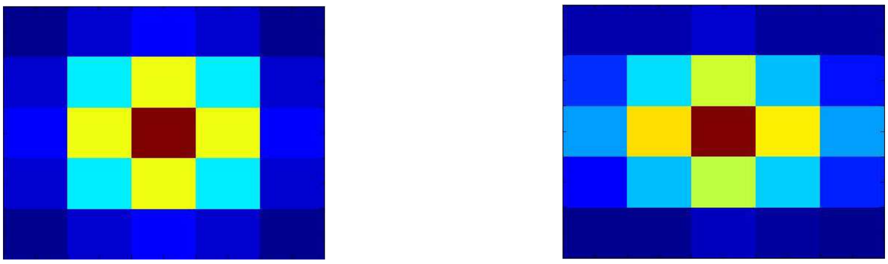


(i) An example of a non blurry denoised frame corrupted by gaussian blur and additive gaussian noise

(ii) The blurred and noisy frame. $PSNR = +21.9dB$

(iii) Deblurred frame using the proposed approach. $PSNR + 26.94dB$

Figure 4.6: Example demonstating the restoration we get when dealing with frames corrupted with the same global blur.



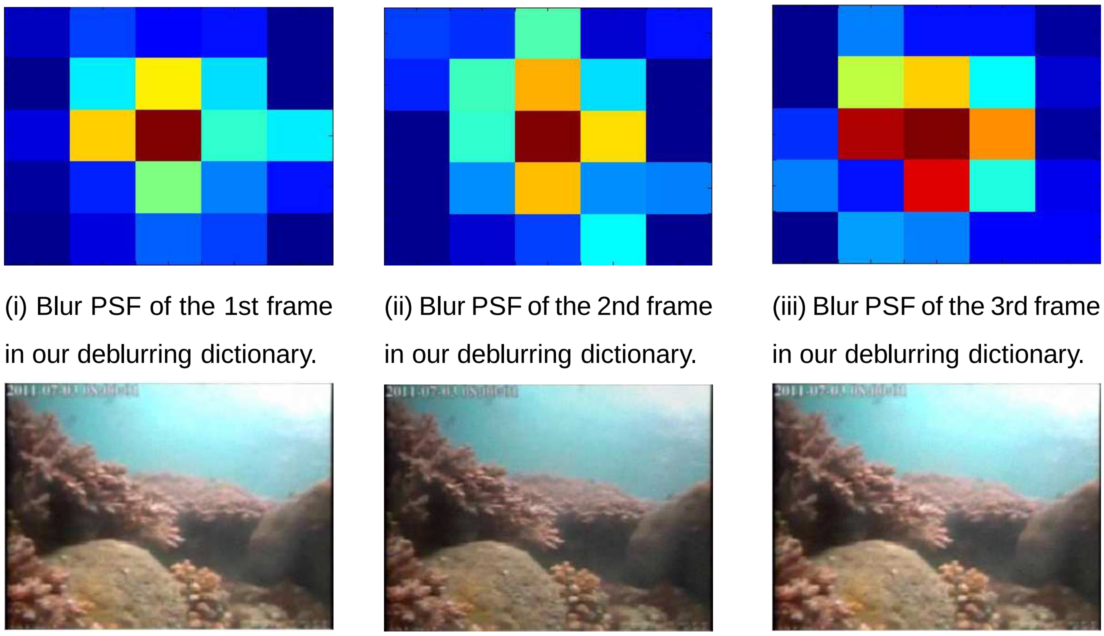
(i) PSF of the blur that was applied to the frames.

(ii) Estimated PSF of the blur using the proposed method.

Figure 4.7: Comparison between the estimated PSF and the actual PSF of the blur.

4.3.1.2 Frames Blurred by Varying Gaussian Blur

In this experiment the Gaussian PSF generated from the previous step was corrupted by additive noise to produce three similar blur PSFs that would be applied to each frame in the sequence. The frames where also corrupted by adding zero-mean Gaussian noise with a variance of 0.0001. This experiment is closer to our assumption that the blur may change slightly from frame to frame. The blur PSFs produced this way can be observed in Figure 4.8 along with the effect they have on the frames of the dictionary.



(i) Blur PSF of the 1st frame in our deblurring dictionary.

(ii) Blur PSF of the 2nd frame in our deblurring dictionary.

(iii) Blur PSF of the 3rd frame in our deblurring dictionary.

(iv) Blurred frame corresponding to the 1st frame in our deblurring dictionary.

(v) Blurred frame corresponding to the 2nd frame in our deblurring dictionary.

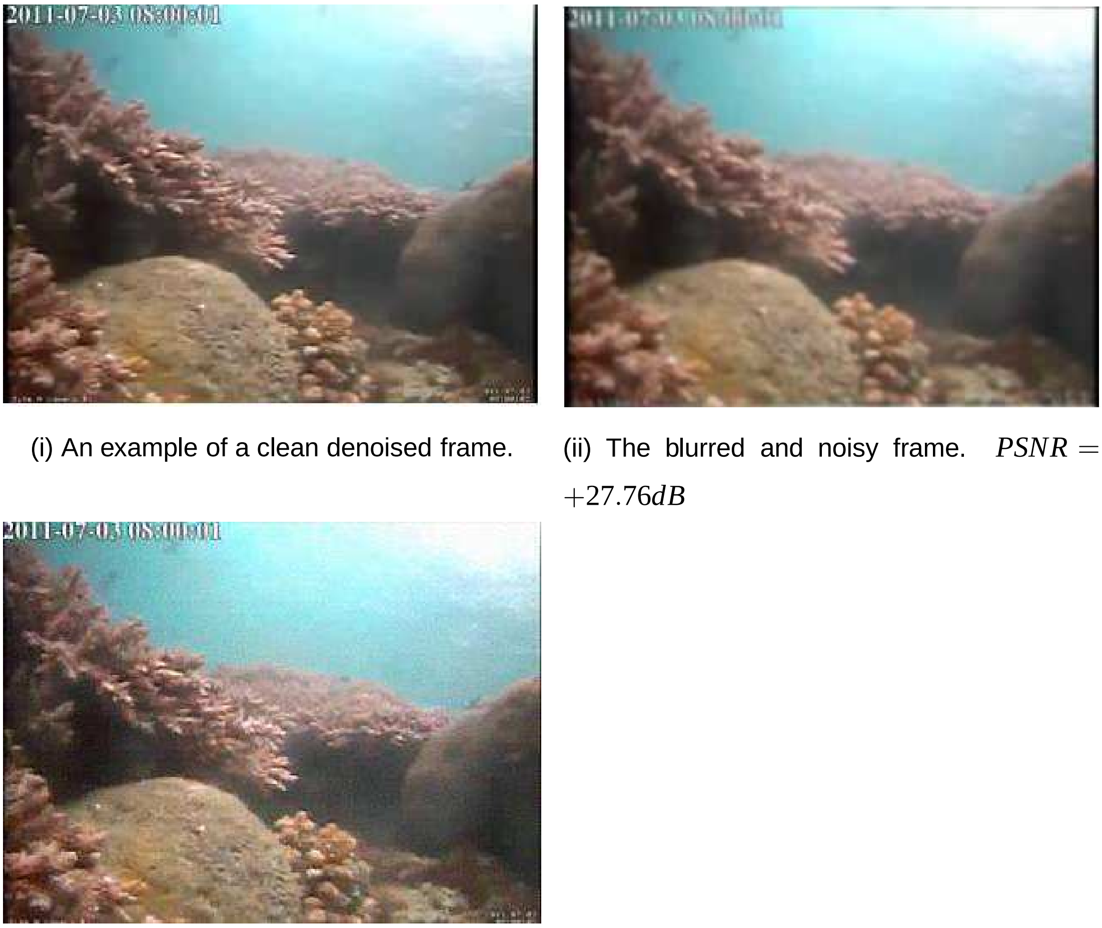
(vi) Blurred frame corresponding to the 3rd frame in our deblurring dictionary.

Figure 4.8: Blur PSFs produced by adding noise to a Gaussian blur PSF. The resulting blurred frames based on the PSFs.

These PSFs are all generated based on the same Gaussian blur but are slightly different between them. This results in frames being blurred in a slightly different manner. An example of how a reconstruction is obtained using a dictionary with the 3 blurred frames shown in Figure 4.9.

The results show that the algorithm managed to reverse the effects of the blur and produce frames that are quite similar to the original. The estimated PSFs that correspond to those in Figure 4.8 are shown in Figure 4.10.

It is evident that the PSFs recovered by the proposed method have captured the

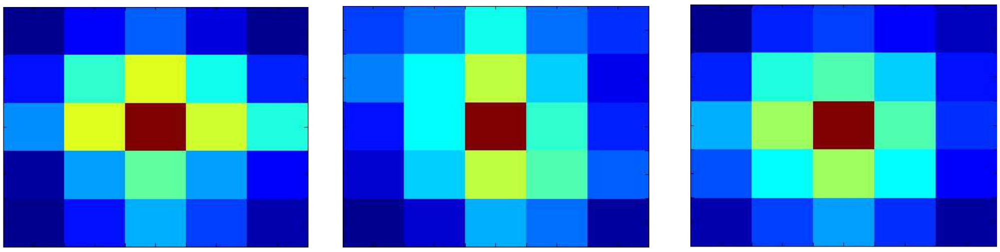


(i) An example of a clean denoised frame.

(ii) The blurred and noisy frame. $PSNR = +27.76dB$

(iii) Deblurred frame using the proposed approach. $PSNR + 31.19dB$

Figure 4.9: Example demonstating the restoration we get when dealing with frames corrupted slightly varying global blur.



(i) Estimated PSF for the blur of the 1st frame in our deblurring dictionary.

(ii) Estimated PSF for the blur of the 2nd frame in our deblurring dictionary.

(iii) Estimated PSF for the blur of the 3rd frame in our deblurring dictionary.

Figure 4.10: Blur PSFs estimated by the proposed method.

structure of the original PSFs well. The PSNR for each PSF estimate was found to be around $35dB$. Which means that each PSF was quite accurately estimated.

4.3.2 Experiments with Artificial Local Blur

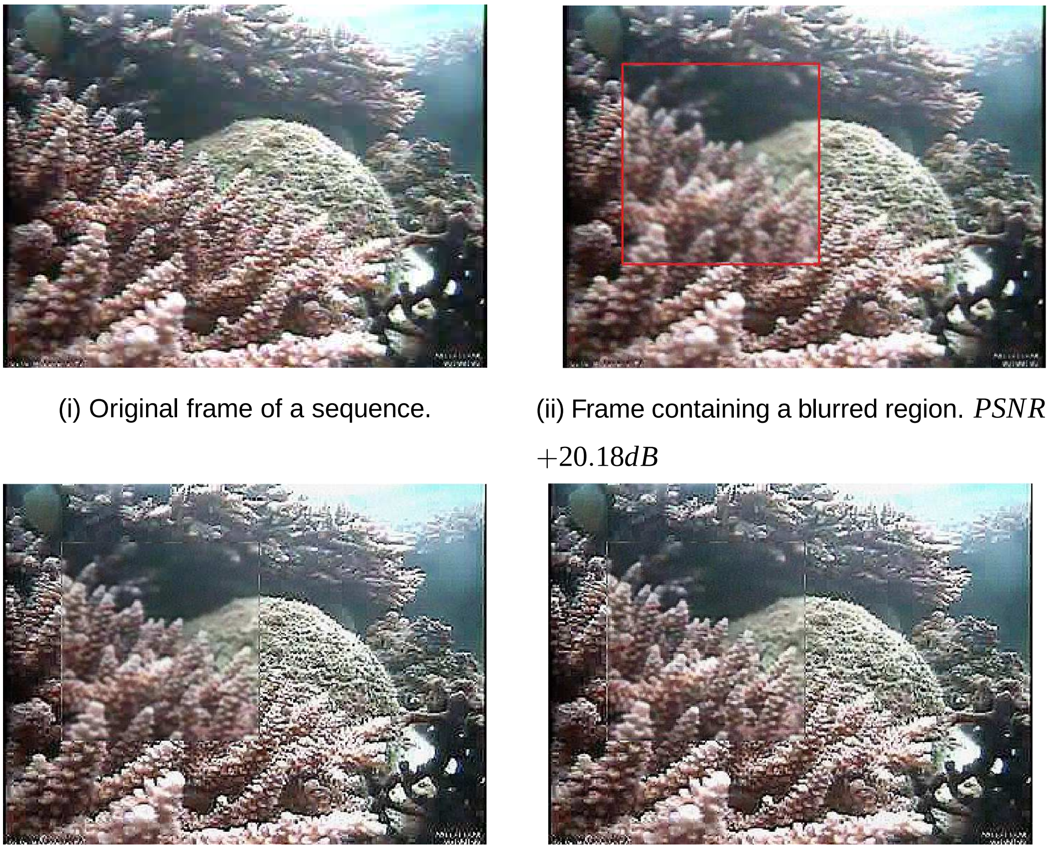
A problem with the underwater recordings is that sometimes a larger amount of dirt gathers only on specific areas of the camera. This causes a strong local blur (similar to fog) in some parts of the frame. The fact that there are clear patches as well as blurry patches in the same frame makes it hard for deblurring algorithms to estimate a single blur PSF for the whole image. For this reason it was decided to experiment using a patch based variation of the proposed multi-frame deblurring method. The patches used for the following experiments where 80 by 80 pixels big.

4.3.2.1 Local Gaussian Blur

In this experiment an area in the frames was artificially blurred using gaussian PSFs and then also corrupted by zero-mean gaussian noise with a variance of 0.0001. The area that was affected by the local blur was chosen to be rectangular for simplicity, although in real data the local blurs can have any shape. The results of the multi-frame method used on patches as well as on the whole image can be seen in Figure 4.11.

From visual inspection the two methods seem to have similar results. The PSNR for the patch-based approach is slightly improved from the whole frame approach. Both of the methods have also sharpened the parts of the image outside the patch. The patch-based method has managed to deal a bit more with the blur inside the patch. This reason behind this improvement can be understood when looking at the PSFs shown in Figure 4.12.

As we can see the frame based approach tries to find a single PSF that can account for the degradation and recovers a PSF that is not similar to the Gaussian PSF of the blur. The patch-based approach however can recover the structure of the gaussian blur for the patch where it belongs. The PSF for a patch not containing blur is closer to a Dirac function, which we were expecting, since the frame is reasonably clear at that region. Despite the effective identification of the structure of the PSF the patch-based approach can't correctly estimate the coefficients of the blurring window and as a consequence doesn't perform much better than the classical approach. This most likely occurs because there is not enough high frequency information present in smaller patches for the deblurring problem to be effectively solved.



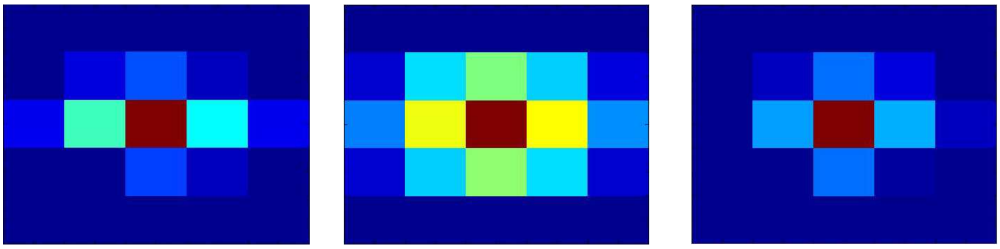
(i) Original frame of a sequence.

(ii) Frame containing a blurred region. $PSNR = +20.18dB$

(iii) Deblurred frame using proposed method on the entire frames.$PSNR = +20.75dB$

(iv) Deblurred frame using proposed method on patches of the image.$PSNR = +21.53dB$

Figure 4.11: The results obtained from using the multiframe algorithm on patches instead of whole images



(i) Blur PSF estimated by the frame-based approach for the 2nd patch in the dictionary.

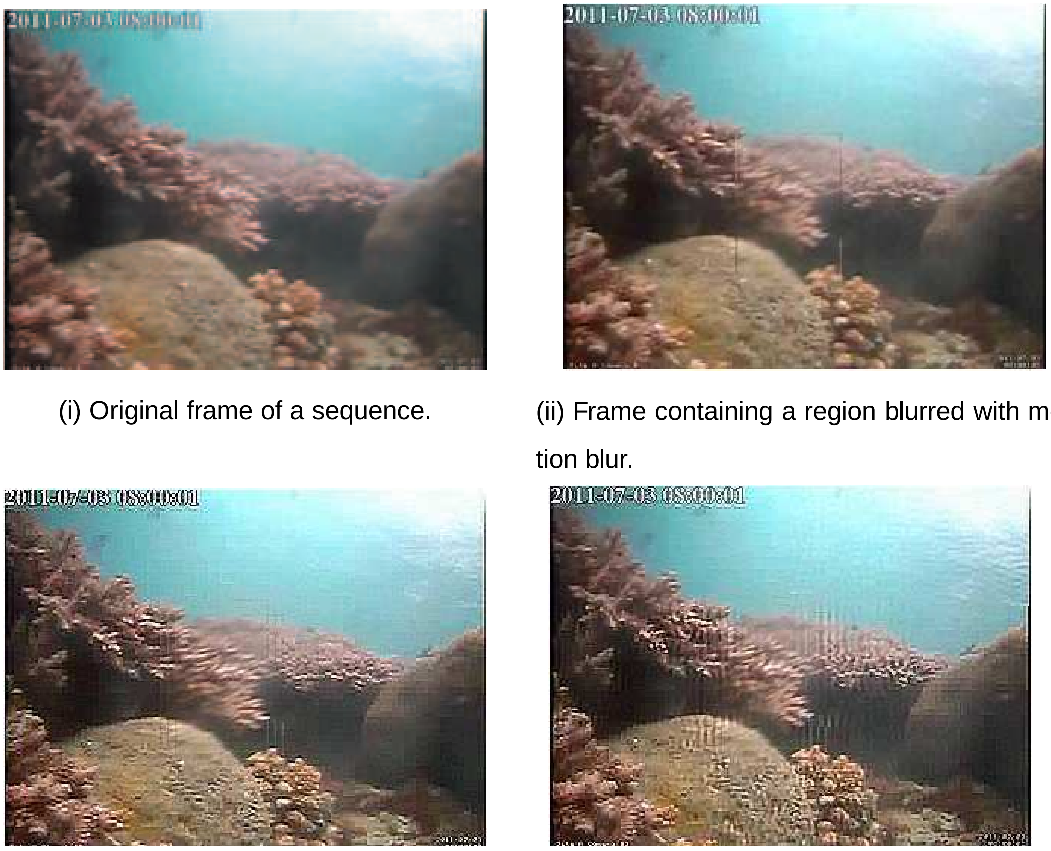
(ii) Blur PSF estimated by the patch-based for a patch where the blur is present

(iii) Blur PSF estimated by the patch-based for a patch where the blur is not present

Figure 4.12: Blur PSFs estimated by the proposed method.

4.3.2.2 Local Motion Blur

It is assumed that sometimes blur in frames can occur due to some swaying motion of the scene. For this reason an experiment with local motion blur was performed. The motion is assumed to change direction between frames. To simulate this frames were locally corrupted by a motion blur whose direction angle varied from frame to frame. An example of a frame from the dictionary that is corrupted by motion blur as well as the reconstruction obtained via the frame-based and patch-based deblurring algorithm are shown in Figure 4.13.



(i) Original frame of a sequence.

(ii) Frame containing a region blurred with motion blur.

(iii) Deblurred frame using proposed method on the entire frames.

(iv) Deblurred frame using proposed method on patches of the image.

Figure 4.13: The results of deblurring on frames corrupted by local motion blur

Both the reconstructions of the frame-based and patch-based algorithm shown seem clearer than the blurred frame. The patch-based approach has achieved a sharper result, although its reconstruction suffers from some artifacts.

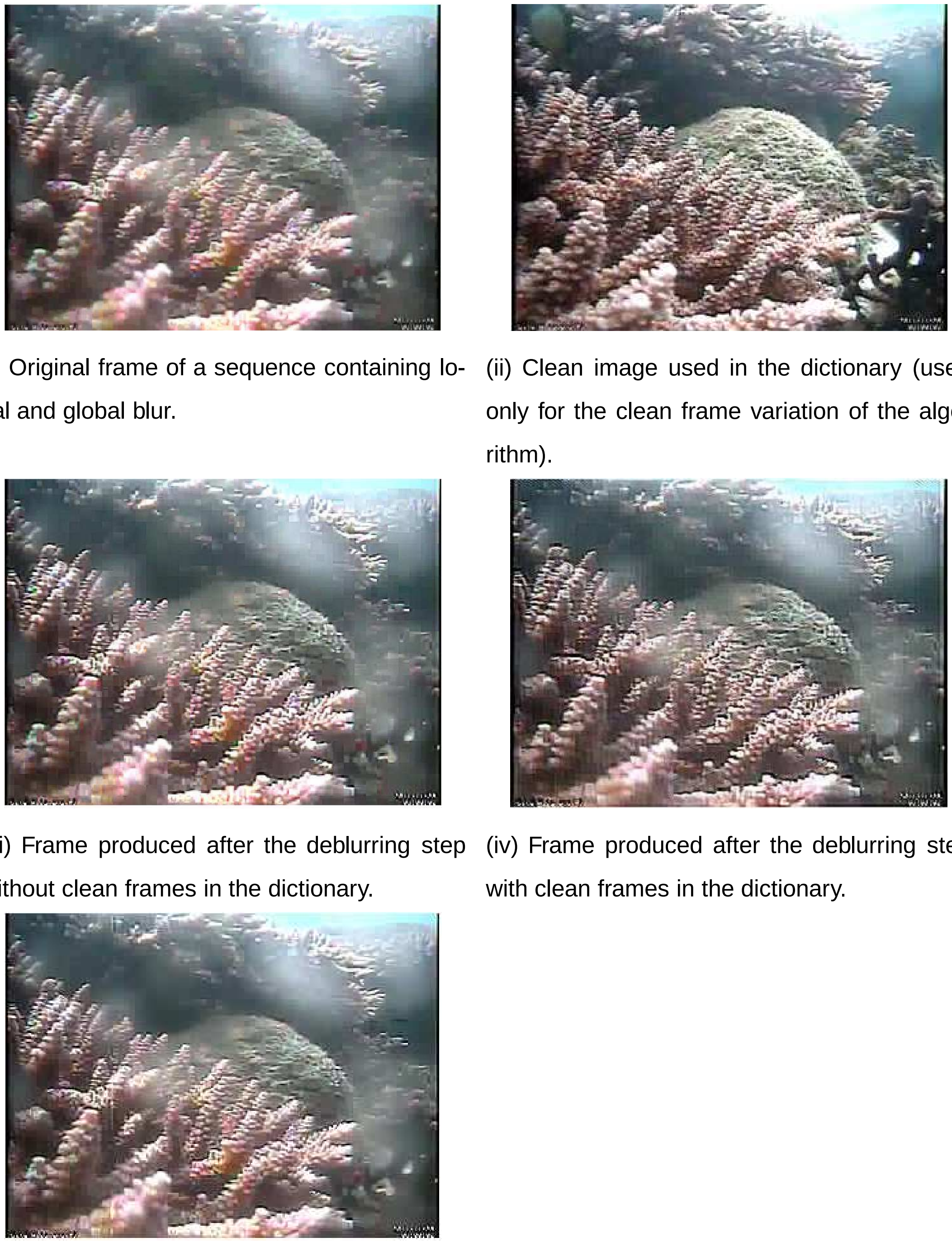
4.4 Experiments with Real Data

The whole algorithm containing both the denoising and deblurring steps was tested on blurry sequences recorded by the underwater camera. In the following experiment the original sequence is not available so the quality of the reconstruction was left for the reader to decide. The basic algorithm proposed in this thesis as well as its proposed variations will be tested on the real data. Therefore, the frame-based as well as the patch-based variation will be assessed. In addition for the frame based algorithm we will examine two cases. In the first case the deblurring stage uses a dictionary containing only the 3 recent blurry frames. In the second case the deblurring stage uses a dictionary containing the 3 recent blurry frames and a frame taken from when the camera was first cleaned. The experiments where conducted on a sequence of 100 blurred frames from our database. The blurs that are present in this example sequence were both local and global blurs. The results of the variations of our video restoration method (patch-based, using clean frames) can be shown in Figure 4.14.

The deblurring algorithms seem to be able to deal with the global blur of the scene. The frame-based approaches can't deal with the local blur in the image as expected. However, the patch based algorithm also struggles with the local blur. A survey was conducted with 11 people out of which 7 preferred the frame-based reconstruction that uses the clean frames and 4 preferred the frame-based approach that doesn't.

The best result was obtained for the method that uses the clean frames. The reasons behind its success can be understood by looking at the point spread functions that where estimated for a clean and blurry frame. These along with the estimated PSF obtained without using clean frames in the dictionary are shown in Figure 4.15.

The first thing that must be noted is that the algorithm estimates a PSF for the clean frame that greatly resembles a Dirac function. This is exactly what we were hoping for because it means that it correctly detects that there is almost no blur in that frame. This also means that the the algorithm will attempt to find PSFs for the blurry frames that through non-blind deconvolution with their frames will produce images that are close to the clean frame. As we can see from Figure 4.15 the variation using the clean frame has managed to retrieve a much more complex PSF that the variation using only blurred frames. Inserting the clean image in the dictionary acts as an added constraint to the minimization problem. For this reason we were able to relax the regularization factor γ of the term $R(h_1, \cdots h_m)$ in the optimization problem in (3.6). Thus a more complex and accurate estimation for the PSFs is obtained that still produces valid results.



(i) Original frame of a sequence containing local and global blur.

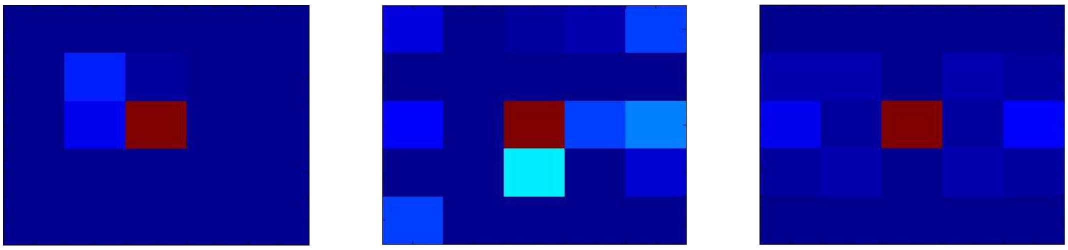
(ii) Clean image used in the dictionary (used only for the clean frame variation of the algorithm).

(iii) Frame produced after the deblurring step without clean frames in the dictionary.

(iv) Frame produced after the deblurring step with clean frames in the dictionary.

(v) Frame produced using the patch-based deblurring step.

Figure 4.14: The results of deblurring on real blurry frames.



(i) Estimated point spread function corresponding to the clean frame. The dictionary used contains clean and blurry frames

(ii) Estimated point spread function corresponding to the blurry frame. The dictionary used contains clean and blurry frames.

(iii) Estimated point spread function corresponding to the blurry frame. The dictionary used contains only blurry frames

Figure 4.15: The estimated point spread functions of for the clean and blurry frames.

Chapter 5

Conclusions and Discussion

5.1 Conclusions

In the previous section the developed method was tested through a series of experiments on data with artificial corruption as well as on real data. The algorithm seems to be able to improve the quality of the frames in most cases. However, it can usually only deal with the global blur caused by floating particles and light diffusion. There are numerous conclusions that can be made based on the results of Chapter 4.

The denoising stage makes the algorithm robust and capable of functioning in cases where the re is noise present in the scene. It is capable of effectively dealing with most types of noise due to the fact that it is a temporal method. The advantages of temporal methods is made evident in the experiments of section 4.2. Despite it being a patch-based algorithm it does not share the two main disadvantages that these algorithms have. Firstly it does not require an exhaustive search in order to group patches together which makes it more suitable for real time applications. Secondly it is able to deal with large amounts of noise as well as “salt and pepper” noise. Its performance is in most cases comparable to applying BM3D on each frame separately without having to perform the second step of Wiener filtering. This means that it does not require knowledge of the variance of the noise, which in most cases is unknown and hard to calculate. The main disadvantage of this method is that it is only useful in the surveillance setting.

The deblurring stage is able to deal with some of the blur in the frames, although it seems to struggle with local blur. Our proposed patch-based variation of the deblurring algorithm can sometimes deal with local blur as shown for the case of artificial local blur of section 4.3.2. Unfortunately, for the frames of the real sequences this problem is still persistent. However, the problem of local blur has not yet been solved by researchers and we believe that a solution might still be possible using a patch-based approach.

Finally, incorporating the clean frames in the deblurring dictionary manages to achieve quite impressive results when dealing with global blur.

5.2 Future Work

After having drawn the above conclusions about the method we believe that there are ways to improve the current algorithm. Firstly, the denoising step could be modified so that it permits overlapping and non-rectangular patches. This will hopefully help in the elimination of the rectangular artifacts. Secondly, the deblurring problem that is dealt with by the multiframe blind deconvolution approach could be redefined so that the blur PSFs $h_1, h_2 \cdots h_m$ are considered similar. This dependence of the PSFs will result in solving an optimization problem for less unknowns making the problem simpler and less ill-posed. In the case where clean frames are added to the dictionary of the deblurring algorithm the optimization problem could be simplified by manually fixing the PSF of the clean frame to be equal to a Dirac function. It is believed that this would also improve the performance of the algorithm.

The patch-based approach could be modified to allow patches of more shapes and sizes that can overlap. In order to accommodate these changes an aggregation method of obtaining pixel estimates should be designed. These changes might be able to resolve some of the issues that the patch-based approach is currently struggling with.

Bibliography

[1] F. J. Sroubek Filip, “Multichannel blind deconvolution of spatially misaligned images,” *Image Processing, IEEE Transactions . . .* , 2005.

[2] R.Schettini and S.Corchs, “Underwater image processing: state of the art of restoration and image enhancement methods,” *. . . Journal on Advances in Signal Processing*, 2010.

[3] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, “Weighted median filters: a tutorial,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 43, no. 3, pp. 157–192, 1996.

[4] A. Buades, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 60 – 65, 2005.

[5] S. Chang, “Adaptive wavelet thresholding for image denoising and compression,” *Image Processing, IEEE . . .* , vol. vol. 9., no. No. 9, pp. 1532–1546, 2000.

[6] H. Choi and R. G. Baraniuk, “Analysis of wavelet-domain Wiener filters,” in *Time-Frequency and Time-Scale Analysis, 1998. Proceedings of the IEEE-SP International Symposium on*, pp. 613–616, 1998.

[7] L. Kaur, “Image denoising using wavelet thresholding,” in *INDIAN CONFERENCE ON COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, AHMEDABAD*, 2002.

[8] K. Dabov and A. Foi, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *Image Processing, IEEE . . .* , vol. 16, no. 8, pp. 2080 – 2095, 2007.

[9] C. Liu and W. T. Freeman, “A high-quality video denoising algorithm based on reliable motion estimation,” *Computer VisionECCV 2010*, 2010.

[10] N. Wiener, “Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications,” *Journal of the American Statistical Association*, 1949.

[11] O. Makhnin, “Image deblurring as an inverse problem Can you read this ?,” 2010.

[12] D. Kundur and D. Hatzinakos, “Blind image deconvolution revisited,” *Signal Processing Magazine, IEEE*, vol. 13, no. 6, pp. 61 – 63, 1996.

[13] D. Kundur and D. Hatzinakos, “Blind image deconvolution,” *Signal Processing Magazine, IEEE*, vol. 13, no. 3, pp. 43 – 64, 1996.

[14] Campisi P. and Egiazarian K., *Blind image deconvolution: theory and applications*. 2007.

[15] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” *. . . Recognition, 2004. ICPR 2004. Proceedings of . . .* , 2004.

[16] Kostadin Dabov and Ro Foi and Karen Egiazarian, “Video denoising by sparse 3D transform-domain collaborative filtering,” *Proc. 15th European Signal Processing . . .* , 2007.

[17] F. J. Sroubek Filip, “Multichannel blind iterative image restoration,” *Image Processing, IEEE Transactions . . .* , 2003.

[18] Z. Wang and A. Bovik, “Mean squared error: Love it or leave it? A new look at signal fidelity measures,” *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98–117, 2009.

[19] A. Danielyan, “BM3D frames and variational image deblurring,” *Image Processing, IEEE . . .* , vol. 21, no. 4, pp. 1715 – 1728, 2011.

[20] A. Jain, *Fundamentals of digital image processing*. 1989.