Fourier Based Pre-Processing For Seeing Through Water

Jerin Geo James
IIT Bombay
jeringeo@cse.iitb.ac.in

Ajit Rajwade
IIT Bombay
ajitvr@cse.iitb.ac.in

Abstract

Consider a scene submerged underneath a fluctuating water surface. Images of such a scene, when acquired from a camera in the air, exhibit significant spatial distortions. In this paper, we present a novel, computationally efficient pre-processing algorithm to correct a significant amount ($\approx 50\%$) of apparent distortion present in video sequences of such a scene. We demonstrate that when the partially restored video output from this stage is given as input to other methods, it significantly improves their performance. This algorithm involves (i) tracking a small number N of salient feature points across the T frames to yield point-trajectories $\{\boldsymbol{q_i} \triangleq \{(x_{it}, y_{it})\}_{t=1}^{T}\}_{i=1}^{N}$, and (ii) using the point-trajectories to infer the deformations at other non-tracked points in every frame. A Fourier decomposition of the N trajectories, followed by a novel Fourier phase-interpolation step, is used to infer deformations at all other points. Our method exploits the inherent spatio-temporal characteristics of the fluctuating water surface to correct non-rigid deformations to a very large extent.
The source code, datasets and supplemental material *can be accessed at [1], [2].*

1. Introduction

In most computer vision applications, the scene being imaged and the imaging sensor (camera) are both located in the same medium (usually air). However there are some applications, where the scene could be located in water but imaged by a camera in the air [16], or vice-versa [3]. In such cases, the images acquired by the camera contain prominent spatial distortions due to the refraction that occurs at the boundary between the two media. Moreover, the water-air boundary can dynamically change its geometry due to external forces such as wind, yielding a dynamic nature to the refraction phenomenon resulting in time-varying non-rigid distortion. Such distortion can adversely affect the performance of typical computer vision algorithms for tracking of objects, object or motion segmentation, object detection, or object recognition. Such tasks arise in applications like surveillance of marine life [12, 18], of shallow water-beds [22], or in ornithological applications such as [14]. Hence, there is motivation to develop algorithms to process the acquired video sequences to remove the spatial distortions.

Previous work in underwater image restoration: This particular problem is relatively unexplored, with only a small-sized body of literature. A large subset of this literature uses some form of optical flow estimation. For example, the classical work in [16] estimates dense optical flow from one frame to another, to trace dense point trajectories. The restoration is performed by undoing the displacements estimated w.r.t. the centroid of the point trajectory at each point. On the other hand, the work in [17] estimates the non-rigid deformation between each frame of the video sequence with an evolving latent image, initialized to be simply the average of all distorted frames. In similar spirit, the work in [11] aligns all video frames to a reference, which is selected to be the least blurred frame. The work in [21] uses PCA to infer a low-rank dictionary to represent non-rigid motion fields. The dictionary is trained on simulated underwater scenes generated by executing the wave equation. The deformation estimation proceeds by first inferring dictionary coefficients.

There also exist approaches which are not pivoted on optical flow. For example, the 'lucky region approach' from [9], [24], [10] and [23] identifies distortion-free patches and mosaics them using graph algorithms. The basic principle is that such distortion-free patches correspond directly to a locally flat portion of the water surface. The technique in [20] frames the restoration problem as a blind deblurring problem, with the average of all video frames used as input. The core theory is that if the water surface is a unidirectional cyclic wave, then the motion blurred average frame can be represented as the convolution of a single blur kernel with a latent clean image. The work in [15] trains a deep neural network to restore *single* distorted underwater images (*not* entire video sequences) by inferring the motion field w.r.t. an unknown clean image automatically. In [13], a set of salient feature points are tracked, and the deformation field is obtained using a compressive sensing framework, by exploiting the Fourier-sparsity of the latent deformation fields.

Overview: In this paper, we present a novel method using simple principles of physics and geometry that exploits the inherent spatio-temporal redundancy of water waves. We model the water surface to be dominantly a superposition of constant-velocity waves, in addition to small local disturbances that get quickly attenuated. This is a very general and widely applicable model. A specific form of this model has been used in [16], in the form of a superposition of *sinusoidal* waves. The model in this paper is more general than that in [16]. In our method, we track some N salient feature points across the T video frames to yield point-trajectories $\{\boldsymbol{q_i} \triangleq \{(x_{it}, y_{it})\}_{t=1}^{T}\}_{i=1}^{N}$. The deformations at all other points in every frame are then interpolated in a novel manner. The deformation-interpolation is performed using a Fourier decomposition of the so-called 'displacement-trajectories' derived from point-trajectories, followed by a *phase-interpolation* step. We observe in real video sequences, that this step is able to correct for a very large amount ($\approx 50\%$) of the undesired motion. Extensive comparisons on real videos show that our method is efficient and advances the performance of the state of the art methods.

Organization: The main theory (assumptions and algorithm) for our method is explained in Section 2. The datasets and experiments are described in Section 3, followed by a discussion and conclusion in Section 4.

2. Assumptions and Main Algorithm

In this section, we begin by describing the main computational task with greater precision and state the various assumptions made.

2.1. Assumptions for Image Formation

We consider a stationary single-plane scene being imaged. We assume that the scene is present below a fluctuating water surface which is shallow and devoid of turbidity. A video sequence of the scene is acquired by a camera which is located in air. The optical axis of the camera is aimed vertically downwards at right angles to the plane containing the scene. Each image (or video-frame) can then be considered to be acquired under orthographic projection. The video-frames are assumed to be relatively free of motion-blur as well as reflection artifacts off the water surface. All these assumptions are valid in a practical setup, as we shall demonstrate from our results on real acquisitions in Section 3. These assumptions are also common in existing literature such as in [21, 20, 16, 17], though [20] expressly models the motion blur for a *specific* unidirectional wave model. Let $\bar{J}$ be the image acquired by the camera if the water surface were perfectly still. Such an image is devoid of spatial distortions. Now, the distorted image J of the same scene acquired given a wavy water surface, can be expressed in the form:

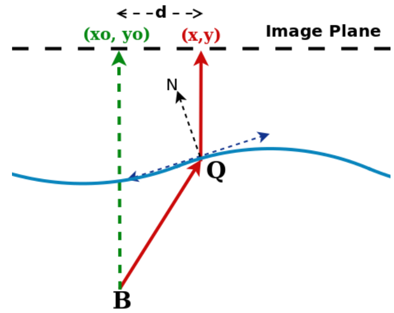

Figure 1. Refractive image formation at a wavy water surface

$$J(\bar{x}, \bar{y}, t) = \bar{J}(\bar{x} + d_x(\bar{x}, \bar{y}, t), \bar{y} + d_y(\bar{x}, \bar{y}, t)), \quad (1)$$

where $(d_x(\bar{x}, \bar{y}, t), d_y(\bar{x}, \bar{y}, t))$ is the displacement at the point $(\bar{x}, \bar{y})$ located in the *undistorted* image $\bar{J}$, at time t. Let $z(x, y, t)$ be the dynamic height of the water surface at time t, above the plane containing the scene. Let $(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})$ be the height-field derivatives at time t, at point Q on the water surface, seen in the ray diagram in Fig.1. Q is the point on the water surface where the ray from point B in the water gets refracted into the air and forms an image at point (x, y) on the camera plane at time t, even though the undistorted coordinates are $(\bar{x}, \bar{y})$. Let μ be the refractive index of water. Then prior work [16] has proved that:

$$(d_x(\bar{x}, \bar{y}, t), d_y(\bar{x}, \bar{y}, t)) = h(1 - 1/\mu) \frac{(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})}{\sqrt{1 + z_x^2 + z_y^2}} \quad (2)$$

$$\approx h(1 - 1/\mu)(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}), \quad (3)$$

where the approximation is valid if $(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 \ll 1$, i.e. for water waves with small slopes. The main task is to obtain $\bar{J}(\bar{x}, \bar{y})$ for all $(\bar{x}, \bar{y})$ with $\{J(:, :, t)\}_{t=1}^{T}$ as input.

2.2. Water Surface Models

In our work, we model the wavy water surface dominantly as a mixture of K constant-velocity unidirectional waves. This is common in situations where waves are generated by more than one disturbance to the still water surface. In addition, the water surface may have small local residual motion that cannot be easily modelled. Such residual motion is expected to be corrected after our Fourier-based pre-processing stage from Alg. 1. Mathematically, the dominant functional form we have is:

$$z(x, y, t) = \sum_{k=1}^{K} \alpha_k g_k(\omega_{tk} t + \omega_{xk} x + \omega_{yk} y + \zeta_k), \quad (4)$$

where α_k is the amplitude of the k^{th} wave, $\omega_{tk}, \omega_{xk}, \omega_{yk}$ stand for its frequency in the t, x, y axes respectively, and ζ_k

stands for a constant phase-lag. The functions $\{g_k\}_{k=1}^{K}$ are any periodic (not necessarily sinusoidal), real-valued and differentiable functions, and they may or may not have the same functional form as each other. Our method does not require any estimate or prior knowledge of K.

2.3. Main Algorithm

Our algorithm consists of many different steps described in the following sub-sections, presented together in Alg. 1. The guiding principle behind it can be described as follows. If the water surface consisted of a single periodic wave, then all point-trajectories (defined precisely below) would be cyclic shifts of one another. Due to this, the phase of their Fourier transforms would form a single plane as defined in Eqn.6. Given just a few salient point trajectories, this property can be used to estimate the motion at all other (non-tracked points), and hence remove the undesired apparent motion in the video frames. On the other hand, if the water surface is the superposition of K different waves, then a similar approach can still be used provided the K waves have disjoint supports in the Fourier domain. If their supports are not disjoint (referred to as 'conflating frequencies'), then additional motion correction needs to be performed using typical optical flow methods. In either case, such a Fourier-based method acts as a very efficient pre-processing step to quickly reduce a large percentage of the apparent distortion.

2.3.1 Salient feature point tracking

Similar to the technique in [13], the first step of our method consists of tracking N salient feature points from the first frame, to yield so-called point-trajectories $\{\boldsymbol{q_i} \triangleq \{(x_{it}, y_{it})\}_{t=1}^{T}\}_{i=1}^{N}$. The coordinates (x_{it}, y_{it}) represent the position in frame t of the i^{th} point whose (initially unknown) coordinates in the distortion-free image $\bar{J}$ are denoted as $(\bar{x}_i, \bar{y}_i)$. For salient feature point detection, we rely on a method based on Difference of Gaussians (DoG) used by SURF[5]. While more sophisticated methods exist [4], they are not deemed essential, as we are interested in just a moderate number $N \sim 100$ of such points. Any salient point (x_{i1}, y_{i1}) detected in the first frame was tracked in subsequent frames using the well-known KLT tracker. A few examples of point tracking on real sequences are shown in the supplemental material folder 'Motion_Reduction'. While there clearly exist many more advanced tracking algorithms, we noted that the KLT tracker was sufficient for this application.

2.3.2 Computing displacement trajectories

Each point-trajectory $\boldsymbol{q_i}$ corresponds to the *unknown* point $(\bar{x}_i, \bar{y}_i)$ in $\bar{J}$. We approximate $(\bar{x}_i, \bar{y}_i)$ by $\tilde{x}_i \approx$

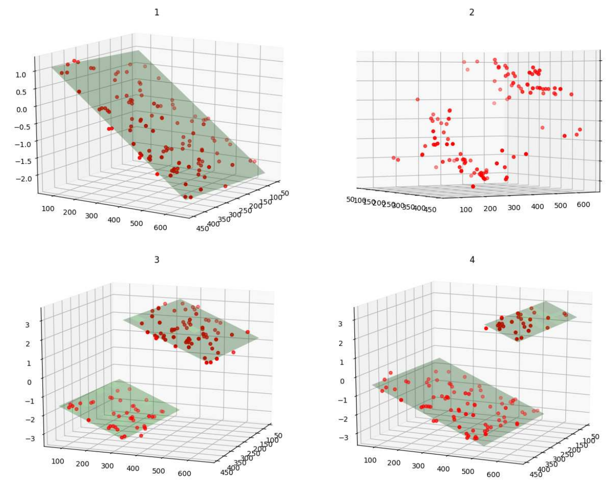

Figure 2. Scatter plot of phases (vs. X,Y) estimated from different displacement trajectories from a real video ('Dices'), and RANSAC-based plane fit. This shows the shift-plane property (phase factor versus x, y) and lack of it (top right sub-figure) for four different frequencies.

$\sum_{t=1}^{T} x_{it}/T, \tilde{y}_i \approx \sum_{t=1}^{T} y_{it}/T$. Although this is an approximation, it is well justified by the assumption that the average of the surface normals $(z_x(x,y,t), z_y(x,y,t))$ across time at any point (x,y) on the water surface, is close to the vertical line $(0,0,1)$ [16]. This is sometimes called the Cox-Munk law [8]. Our experiments with synthetic and real video sequences confirm its validity for even as less as $T \sim 50$ frames. This is partly conveyed by Fig.4, where the image quality metric saturates after $T \sim 50$ frames. Also, an example illustrating the convergence of $(\bar{x}_i, \bar{y}_i)$ is included in the supplemental material. With this, our set of displacements for the i^{th} salient feature point are given as $\boldsymbol{d_i} \triangleq (\boldsymbol{d_{ix}}, \boldsymbol{d_{iy}}) \triangleq \{(x_{it} - \tilde{x}_i, y_{it} - \tilde{y}_i)\}_{t=1}^{T}$. We term these as 'displacement-trajectories', just as in [13].

2.3.3 Fourier decomposition

First, let us consider the case of a single wave, i.e. $K = 1$ in Eqn.4, and $z(x,y,t) = \alpha_1 g_1(\omega_{t1} t + \omega_{x1} x + \omega_{y1} y + \zeta_1)$. We will soon generalize to the case when $K > 1$. The displacements $\boldsymbol{d_i}$ across time at any point $(\bar{x}_i, \bar{y}_i)$ turn out to form a cyclic sequence. This can be understood from Eqn.3 (with or without the small-wave approximation) given the cyclic nature of z. Hence, the respective displacement-trajectories $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ at any two points $(\bar{x}_i, \bar{y}_i)$ and $(\bar{x}_j, \bar{y}_j), i \neq j$, are cyclic shifts (in time) of each other. This shift is equal to the effective distance between the two points covered by the wave, i.e. $(\bar{x}_i - \bar{x}_j, \bar{y}_i - \bar{y}_j) \cdot (\hat{\omega_{x1}}, \hat{\omega_{y1}})$, divided by the wave velocity $\frac{2\pi}{T\sqrt{\omega_{x1}^2 + \omega_{y1}^2}}$. Here $(\hat{\omega_{x1}}, \hat{\omega_{y1}})$ is the unit-norm direction vector of the wave. Since the wave velocity is constant, by the Fourier shift theorem we have

$$\mathcal{F}[\boldsymbol{d_{ix}}](u) = \exp\big(-\frac{\iota 2\pi u(a\Delta_{x,i,j} + b\Delta_{y,i,j})}{T}\big)\mathcal{F}[\boldsymbol{d_{jx}}](u), \tag{5}$$

and likewise for $\boldsymbol{d_{iy}}, \boldsymbol{d_{jy}}$ with the *same* phase factor. Here u is the frequency, $\Delta_{x,i,j} \triangleq \bar{x_j} - \bar{x_i}$, $\Delta_{y,i,j} \triangleq \bar{y_j} - \bar{y_i}$, $\iota \triangleq \sqrt{-1}$, $\mathcal{F}$ is the 1D Fourier operator (applied independently for x and y components), and (a, b) are constants independent of t, x, y but directly proportional to $(\omega_{x1}, \omega_{y1})$. Hence the Fourier domain phase shifts between $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ at frequency u are given as: $\phi_{u,j} - \phi_{u,i} = \big(2\pi u(a\Delta_{x,i,j} + b\Delta_{y,i,j})/T\big)\%2\pi$, where $\%$ represents the remainder after division (mod). From this expression, we see that the phase factors of the displacement-trajectories $\boldsymbol{d_j}$ for all $j \in \{1, ..., N\}$ form a plane of the following form:

$$\phi_{u,j} = \big(2\pi u(a\bar{x_j} + b\bar{y_j} + c)/T\big)\%2\pi. \tag{6}$$

The unknown parameters are (a, b, c) where c is a constant offset, $\phi_{u,j}$ is the dependent variable, and $\bar{x_j}, \bar{y_j}$ are independent variables. We hereafter refer to this as the **shift-plane property**, illustrated in the Fig.2 for the $K > 1$ case (see also Sec. 2.3.5). Although we refer to it as a plane, it is strictly speaking a small number of parallel planes, due to the $\%$ operator in Eqn.6. Given $N \geq 3$ points, the plane parameters can be estimated using a least squares fit that minimizes $\sum_{j=1,j\neq i}^{N}\sum_{u=0}^{T-1}\big(\phi_{u,j} - \big(2\pi u(a\bar{x_j} + b\bar{y_j} + c)/T\big)\%2\pi\big)^2$. Of course, one usually prefers a larger N as well as a RANSAC-based robust plane fit to handle errors in the displacement-trajectories (that may arise due to errors in point-trajectories). Given the estimates of a, b, c, we can obtain the displacement-trajectory at any point $(\bar{x_m}, \bar{y_m})$ in the image domain, *including points which were not tracked*, by (i) using Eqn.6 to find $\phi_{u,m}$, and (ii) using Eqn.5 to determine $\boldsymbol{d_m}$ treating $\boldsymbol{d_j}$ as reference, without loss of generality. Thus, our algorithm makes use of inherent *spatio-temporal properties of water waves* to interpolate the deformation field for the whole image, starting with a small number of point-trajectories. In contrast, standard optical flow algorithms are not designed to exploit this information and only use *local* spatial regularizers of different types, or (much less commonly) *local* temporal regularization as well [6]. However, our method uses *global* properties of the water waves. A sample result of our technique on a synthetic single wave dataset is shown in the *supplemental material*. This geometric treatment however is no longer applicable when $K > 1$, which is the more general model. In such a case, even though the displacement-trajectories caused due to constituent waves are shifted versions of each other, the superimposed displacement-trajectories are no longer shifts of each other. That is, the shift-plane property is violated. To deal with this issue, we perform a Fourier decomposition of each displacement-trajectory $\boldsymbol{d_i}$, given as follows:

$$\boldsymbol{d_{ix}} = \sum_{u=0}^{T-1}\beta_{u,i,x}\boldsymbol{f_u}; \boldsymbol{d_{iy}} = \sum_{u=0}^{T-1}\beta_{u,i,y}\boldsymbol{f_u}, \tag{7}$$

where $\boldsymbol{f_u}$ is the $T \times 1$ Fourier basis vector at frequency u, and $\beta_{u,i,x} = |\beta_{u,i,x}|\angle\phi_{u,i}, \beta_{u,i,y} = |\beta_{u,i,y}|\angle\phi_{u,i}$ are the corresponding complex-valued (scalar) Fourier coefficients[1]. Note that all K constituent waves in Eqn.4 contribute to $\beta_{u,i,x}, \beta_{u,i,y}$ for any u, i. Now consider the ideal case when the *dominant* Fourier components of the K constituent waves in Eqn.4 have disjoint support in the frequency domain. In such a setting, all the supports will obey **shift-plane property**. Hence, given a frequency u, only one of the K waves (say the l^{th} wave) has a significant contribution to $\beta_{u,i,x}, \beta_{u,i,y}$ and other waves have a relatively minor contribution. For a different frequency $\tilde{u}$, some other wave (say the $\tilde{l}^{\text{th}}$ wave) could be the sole major contributor. We term this the **'Fourier separation' property** (FSP). For any given u, the signals $\{\beta_{u,i,x}\boldsymbol{f_u}\}_{i=1}^{N}$ denote the contribution of frequency u, i.e. dominantly *only one* of the K waves, to $\boldsymbol{d_{ix}}$ (likewise for y). As per FSP, for a fixed u, each of these signals are shifted versions of each other, on the lines of the $K = 1$ formulation. Hence the phase factors $\{\phi_{u,i}\}_{i=1}^{N}$ of the Fourier coefficients $\{(\beta_{u,i,x}, \beta_{u,i,y})\}_{i=1}^{N}$ lie close to a planar surface of the following form:

$$\phi_{u,i} = (2\pi u(a_u\bar{x_i} + b_u\bar{y_i} + c_u)/T)\%2\pi, \tag{8}$$

with unknown plane parameters a_u, b_u, c_u. For different frequencies, the phase factors will lie close to different planar surfaces (hence the subscript u in the parameters a_u, b_u, c_u). The parameters can be determined using RANSAC as explained before. Also due to FSP, the values $\{|\beta_{u,i,x}|\}_{i=1}^{N}$ (i.e. the magnitudes of the Fourier coefficients) are all equal, and can be denoted as $|\beta_{u,x}|$ (likewise for y). In practice, we computed a median value.

2.3.4 Motion correction

For motion correction, first the plane parameters a_u, b_u, c_u are obtained for every u. However for computational efficiency, this is done only for those frequencies that account for 99% of the signal energy. In our experiments, we found that a set $\mathcal{S}$ of just about 15-20 frequencies (out of $T/2$) sufficed for this. Thereafter for every non-tracked point $(\bar{x_m}, \bar{y_m})$, we compute $\phi_{u,m}$ from Eqn.8. Armed with this, the complete trajectory $\boldsymbol{d_m}$ can be approximated as follows:

$$\boldsymbol{d_{m,x}} = \sum_{u\in\mathcal{S}}|\beta_{ux}|\angle\phi_{u,m}\boldsymbol{f_u}; \boldsymbol{d_{m,y}} = \sum_{u\in\mathcal{S}}|\beta_{uy}|\angle\phi_{u,m}\boldsymbol{f_u}. \tag{9}$$

[1] Note that $\beta_{u,i,x}, \beta_{u,i,y}$ have the same phase (cf Eqns. 5, 6) and possibly different magnitudes.

Note that we drop the subscript m in the magnitude of the Fourier coefficient $|\beta_{ux}|, |\beta_{uy}|$, for reasons explained in Sec. 2.3.3. In this manner, using the special spatio-temporal properties of water waves, the displacement-trajectories at all points in the image domain can be interpolated.

2.3.5 Handling conflating frequencies

Our algorithm is able to accurately estimate the displacement-trajectories at all pixels in the image domain from a small set of salient feature point-trajectories, if the FSP is indeed true. However there can certainly arise cases where two or more constituent waves have partly overlapping dominant supports in the Fourier domain. In such a case, there will be a subset of frequencies $\mathcal{C}_f$ from $\{0, 1, ..., T-1\}$ at which the aforementioned phase-shifts will not form a plane - see Fig.2 for a comparison. To detect such 'conflating frequencies', we first perform the least squares plane fit for each frequency u on a subset $\mathcal{T}$ of $\{\boldsymbol{d_i}\}_{i=1}^N$. For each point $(\bar{x_j}, \bar{y_j})$ in $\{1, ..., N\} - \mathcal{T}$, the predicted partial displacement-trajectory is $\boldsymbol{d^u_{j,x}} \triangleq \boldsymbol{f_u}|\beta_{ux}|\angle\phi_{u,j}$ (likewise for y). We consider u to be a conflating frequency if $\boldsymbol{d^u_{j,x}}$ and $\boldsymbol{d^u_{j,y}}$ do not yield a positive correlation with displacement-trajectories in $\{\boldsymbol{d_j}\}_{j=1}^N$ for most $j \in \{1, ..., N\} - \mathcal{T}$.

If the K waves have some conflating frequencies, then the initial motion correction step based on Eqn.9 has to be modified. Instead, we find partial displacement-trajectories for every pixel $(\bar{x_j}, \bar{y_j})$ as follows:

$$\widetilde{\boldsymbol{d_{j,x}}} = \sum_{u \in \mathcal{S} - \mathcal{C}_f} |\beta_{u,x}|\angle\phi_{u,j}\boldsymbol{f_u}; \widetilde{\boldsymbol{d_{j,y}}} = \sum_{u \in \mathcal{S} - \mathcal{C}_f} |\beta_{u,y}|\angle\phi_{u,j}\boldsymbol{f_u}. \tag{10}$$

These partial displacement-trajectories can be used to correct the deformations partially by simply applying the reverse deformation field to every frame. We have observed that the partial displacement-trajectories (obtained via the Fourier stage) account for $\approx 50\%$ of the original motion in a median sense. Details about the quantification of reduction in motion are explained in Sec. 3.2.1.

2.3.6 Comments about our algorithm

Our Fourier-based method acts as a geometrically- and physically-motivated initial step for further distortion removal by other techniques. As we shall further demonstrate in Section 3, for videos with large motion, state of the art techniques by themselves are unable to yield results of the same quality without initial motion correction with the Fourier-based method.

Input : Distorted video J
Output: Restored image $\tilde{J}$

1. Track N feature points to obtain point-trajectories $\{\boldsymbol{q_i}\}_{i=1}^N$ as per Sec. 2.3.1.
2. Compute displacement trajectories $\{\boldsymbol{d_i}\}_{i=1}^N$ as per Sec. 2.3.2.
3. For each $\boldsymbol{d_i}$, compute Fourier decomposition as per Eqn.7 as per Sec. 2.3.3.
4. For every u, perform RANSAC-based plane fitting to the phase factors $\{\phi_{u,i}\}_{i=1}^N$ of the Fourier coefficients from the previous step as per Eqn.8.
5. Identify non-conflating frequencies, and compute the partial displacement-trajectories using Eqn.10 in Sec. 2.3.5.
6. Perform initial motion correction from the partial trajectories to get an intermediate restored video.
7. Pass this partially restored video as input to other methods, which will yield restored image $\tilde{J}$.

Algorithm 1: Algorithm to Restore Video

Since the method uses RANSAC-based linear interpolation, it is robust to the presence of moderate levels of outliers in the form of reflection or blur. This is because we are able to interpolate the optical flow (at least partially) in all such places based on physical wave properties. We note that our algorithm does *not* break down even if the Fourier separation property is not obeyed for a few conflating frequencies. This is because we are automatically able to detect the conflating frequencies and do not use them for motion correction (Eqn.10). In such cases, we cannot obtain the full deformation from Sec. 3.2.1 and Eqn.10.

It is to be noted that our method is very different from the bispectral approach in [24] which chooses 'lucky' (i.e. least distorted) patches, by comparing to a mean template. In that method, the Fourier transform is computed locally on small patches in the spatial domain for finding similarity with corresponding patches from a mean image. On the other hand, our Fourier decomposition is temporal. The idea of dense optical flow interpolation (not specific to underwater scenes) from a sparse set of feature point correspondences has been proposed in the so-called EpicFlow technique [19]. The interpolation uses non-parametric kernel regression or a locally affine method. However our method uses physical properties of water waves and also considers temporal aspects of optical flow, which is missing in EpicFlow.

Lastly, our approach is also significantly different from [13]. There the entire spatio-temporal displacement vector field, represented as a 3D complex valued signal $\boldsymbol{d}(x, y, t) = \boldsymbol{d_x}(x, y, t) + \iota\boldsymbol{d_y}(x, y, t)$, is considered Fourier-sparse and sampled by means of salient feature point tracking. To be effective, it typically requires a larger number of point-trajectories. On the other hand, our method considers independent Fourier decompositions of individ-

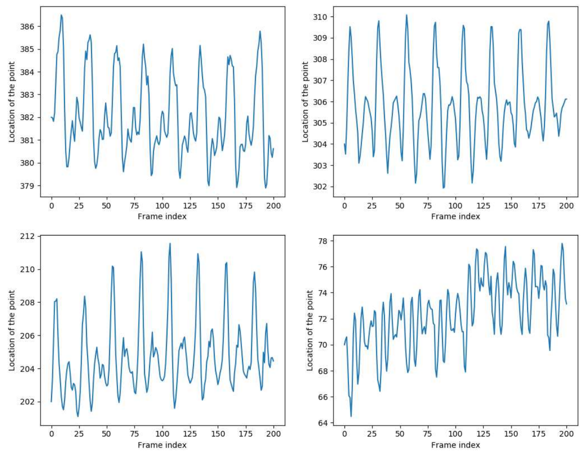

Figure 3. Point-trajectories at four different salient points in a real video sequence. As mentioned in 3.1, this verifies that the water waves are not unidirectional

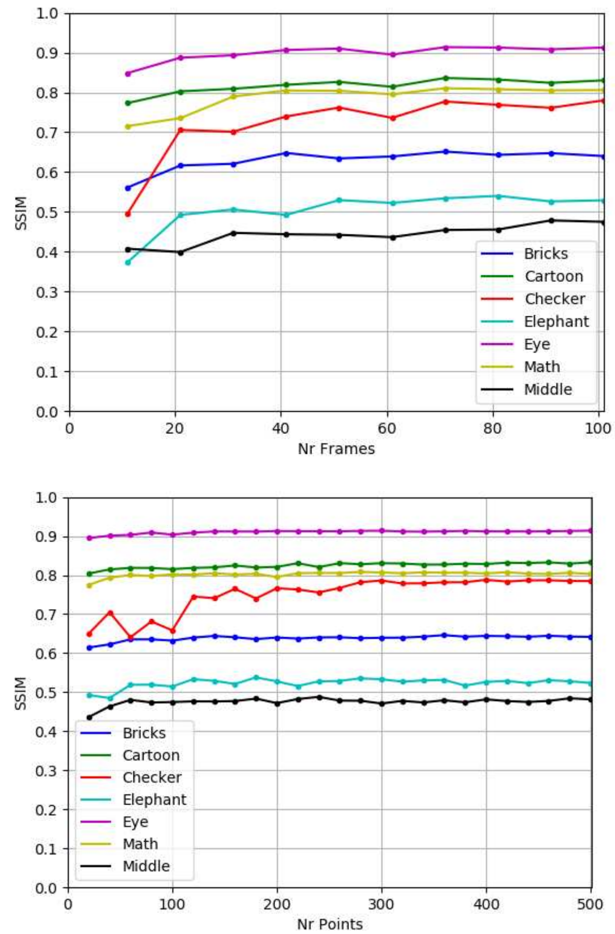

Figure 4. Effect of increase in number of frames T (top) and number of salient points N (bottom) on restoration performance for Fourier method. Notice that the SSIM values get saturated after a small T and N

ual point- or displacement-trajectories, and can work with a smaller number of trajectories.

3. Experimental Results

In this section, we present our results on two datasets of real video sequences, gathered from different sources. All image and video results are available in the *supplemental material*.

3.1. Description of datasets

We demonstrate our algorithm on two sets of real video sequences: **Real1** initially used in [13], and **Real2** initially used in [21]. **Real1** contains real video sequences (of size $\sim 700 \times 512 \times 101$ with a 50 fps camera) of laminated posters kept at the bottom of a water-tank in a 'wave-flume', where waves were generated using paddles. The sequences showed distortions that could not have emerged from single cyclic waves. An example of this can be seen in Fig.3, since the point trajectories at different salient features are *not* cyclic shifts of each other. **Real2** contains three video sequences of size $\sim 300 \times 250 \times 101$, acquired at 125 fps.

3.2. Description of parameters and comparisons

In all the datasets, we tracked around $N = 256$ salient feature points. In rare cases, there were tracking errors leading to trajectory outliers. However, such outliers were filtered out during the RANSAC-based plane fitting step. We evaluate the performance using two measures (1) the reduction in the amount of non-rigid distortions after Fourier stage and (2) improvement in recovered image quality (measured by SSIM and NMI) when Fourier method is used as pre-processing step. Both these measures are explained in the following subsections respectively.

3.2.1 Motion reduction

This quantity indicates the percentage of the distortion estimated (and hence removed) by the Fourier stage. It is calculated as follows: (i) The Fourier interpolation step is performed using displacement trajectories at a set of N points which we denote as $\mathcal{P}_1$. We obtain the displacement trajectories at some N_2 salient feature points, $\{\boldsymbol{d_j}\}_{j=1}^{N_2}$ at some N_2 salient feature points, which form a set $\mathcal{P}_2$ which is *disjoint* from $\mathcal{P}_1$. (ii) We estimate the displacement trajectories $\{\hat{\boldsymbol{d}}_{\boldsymbol{j}}\}_{j=1}^{N_2}$ at locations in $\mathcal{P}_2$ using the Fourier model, performing interpolation via Alg.1 from displacement trajectories at points *only* in $\mathcal{P}_1$ without using those in $\mathcal{P}_2$. Then, we compute the measure of the motion reduction given

	FM					LWB		FM + LWB			SBR		FM+SBR	
	Time	**MR (%)**	**NMI**	**SSIM**		**NMI**	**SSIM**	**NMI**	**SSIM**		**NMI**	**SSIM**	**NMI**	**SSIM**
Real1														
Cartoon	1m 42s	54.91%	1.164	0.848		1.152	0.836	1.179	**0.870**		1.173	0.843	1.232	**0.890**
Checker	2m 3s	35.53%	1.166	0.809		1.105	0.660	1.164	**0.845**		1.158	0.791	1.186	**0.824**
Dices	1m 36s	47.65%	1.109	0.814		1.086	0.783	1.132	**0.869**		1.100	0.758	1.154	**0.876**
Bricks	1m 35s	54.56%	1.119	0.699		1.118	0.673	1.140	**0.775**		1.128	0.686	1.159	**0.770**
Elephant	1m 40s	44.70%	1.081	0.589		1.068	0.584	1.093	**0.699**		1.075	0.516	1.119	**0.724**
Eye	1m 41s	58.95%	1.203	0.915		1.155	0.903	1.209	**0.940**		1.179	0.913	1.265	**0.941**
Math	1m 22s	62.99%	1.106	0.816		1.067	0.766	1.141	**0.885**		1.100	0.841	1.163	**0.857**
Real2														
Middle	1m 12s	40.03%	1.113	0.586		1.163	0.761	1.171	**0.815**		1.189	0.782	1.187	0.775
Small	0m 58s	29.47%	1.118	0.505		1.151	0.688	1.144	**0.704**		1.153	0.741	1.142	0.654
Tiny	1m 46s	10.05%	1.142	0.587		1.167	0.654	1.157	**0.689**		1.161	0.657	1.154	0.625

Table 1. Comparison of various methods on video sequences w.r.t. Running Time, Motion Reduction, NMI, SSIM. Higher SSIM and NMI are better.

as $MR \triangleq \text{median}_{j\in\{1,\ldots,N_2\}} \|\hat{\boldsymbol{d}}_{\boldsymbol{j}} - \boldsymbol{d}_{\boldsymbol{j}}\|_2 / \|\boldsymbol{d}_{\boldsymbol{j}}\|_2$. Hence, this measure indicates how much of the original motion the Fourier stage is able to predict.

3.2.2 Fourier method as pre-processing stage for other methods

The Fourier Method (FM) predicts a significant amount ($\approx 50\%$) of non-rigid distortions, and hence acts as a desirable pre-processing step before other algorithms for motion reduction can be used. We compare two state of the art methods with and without our Fourier-based pre-processing step, to demonstrate that in almost all cases, the Fourier-based step significantly improves their performance. We demonstrate these results on (1) the two-stage method in [17] consisting of spline-based registration followed by *Robust Principal Component Analysis*[7] (SBR) which is considered state of the art for underwater image restoration; (2) the method from [21] using learned water bases (LWB).

For quality assessment referring to ground truth, we used the following measures: (i) visual inspection of the restored video J_r as well as its mean-frame $\bar{J}_r$, (ii) normalized mutual information (NMI) between $\bar{J}_r$ and $\bar{J}$ (grayscale), where $\bar{J}$ is the ground-truth image representing the undistorted static scene, and (iii) SSIM (grayscale) between $\bar{J}_r$ and $\bar{J}$. All the values were calculated after normalizing the intensities of each image to the range $[0, 1]$. We did not compare with [20] since it is modelled on unidirectional wave motion assumption (whereas we assume more general wave models), and due to unavailability of publicly released code. Likewise, we did not compare with [11] due to unavailability of publicly released code. We did not compare with the deep-learning technique in [15], since it did not perform well in comparison to SBR and LWB. This might be because, the deep-learning technique is designed to do restoration from a single distorted image and does not take into account the extra temporal information available in the video sequences. Please see Table.1 of [13] for the quantitative comparison of [15] w.r.t SBR and LWB. We also did not compare with [13] since it is based on the sparsity of the motion vector field and reducing the magnitude of motion does not alter it's performance much.

3.3. Discussion of results

The numerical results are presented in Table 1. The mean images (post-restoration) for a sample video, restored by various methods, are presented in Fig.5. The *supplemental material* contains results on 10 videos (videos and mean images post-restoration) for all methods. Also, Fig.6 higlights local SSIM errors between the mean image produced by restoration with various methods w.r.t. the ground truth image. The *SSIM Overlay Image* is created in the following manner $0.7 \times \text{RestoredImage} + 0.3 \times (1 - \text{SSIM-Map}) \times$ Red-Color. Such a visualization highlights the low SSIM regions with brighter shades of red color. The figure shows that pre-processing the state of the art methods with Fourier method reduced the structural dissimilarity of the restored image w.r.t the ground truth. Our Fourier method was able to achieve around $\approx 50\%$ motion reduction in a median sense, as indicated by the MR column in the Table 1. Also, the table further conveys that the Fourier based pre-processing stage has increased the recovered image quality for all videos for [21] and 7 out of 10 videos for [17]. Also, in the 3 videos where Fourier did not perform well, preceding SBR with FM improved the image quality at the central regions. However, the overall SSIM value got reduced due to artifacts at the borders. This can be observed in the *SSIM overlay* images inside 'Collage_MeanImages' folder in the supplemental material folder. 4 shows the variation in $SSIM$ wrt number of frames and number of tracked salient feature points. It can be observed that both the plots attain saturation after a small number of points. When it comes to

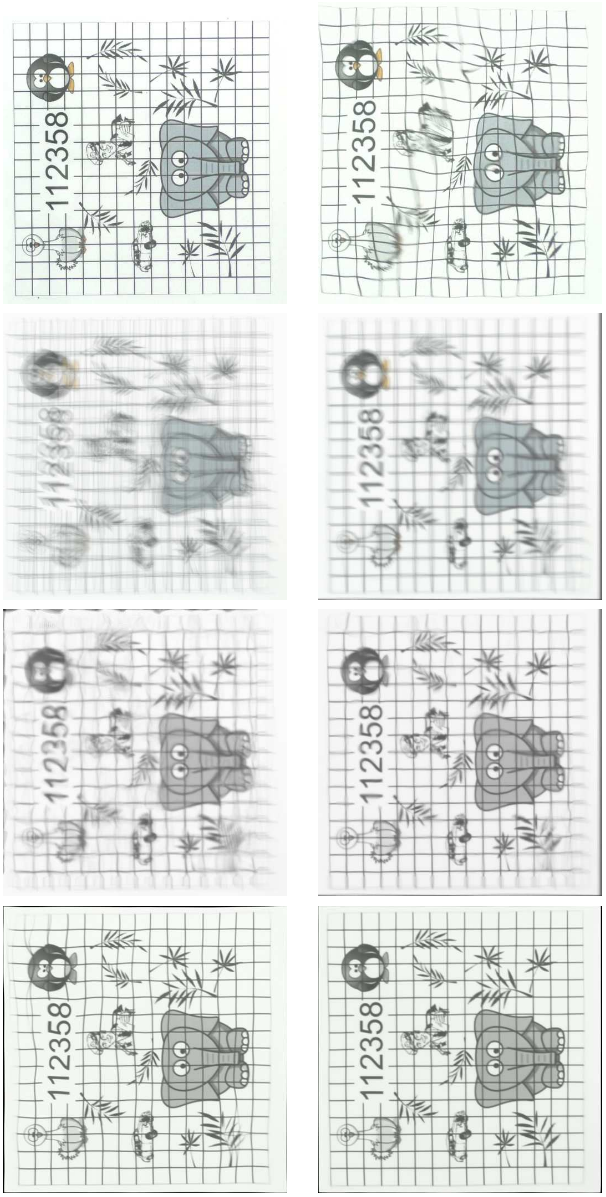

Figure 5. Left to right, top to bottom: mean frame of the video after restoration by the following methods: FM; LWB [21], FM followed by LWB; SBR [17], FM followed by SBR. Zoom into pdf for better view. Notice that geometric distortions in LWB and SBR are corrected when those were preceded by Fourier method.

computational time, SBR and LWB take more than an hour for a single video. As indicated in 1, Fourier based pre-processing step just adds one and a half minutes on average to the processing time and significantly improves the image quality.

4. Conclusion

We have presented a novel method for removal of refractive distortions induced in images of scenes imaged from air but situated underneath a fluctuating water surface, based on a novel usage of Fourier decomposition for interpolating optical flow sequences starting from a very small set of point-trajectories. We have demonstrated that the state of the art methods can be significantly improved with this *computationally inexpensive* pre-processing step.

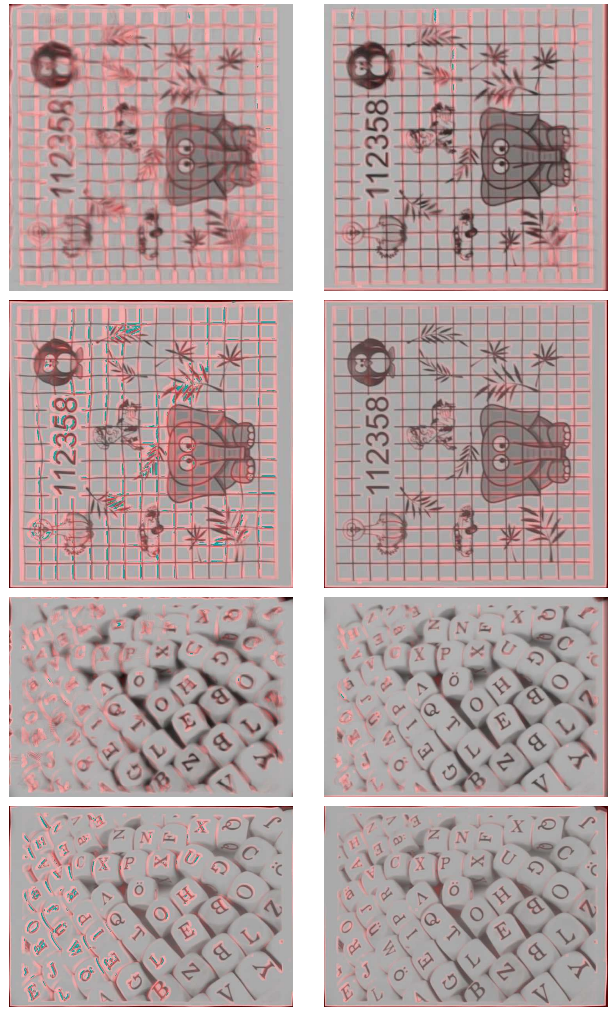

Figure 6. SSIM Overlay : For each of the two set of videos, Left to right, top to bottom order: LWB, FM + LWB, SBR, FM+SBR. More red implies more deviation of the restored image from the ground truth. Notice that pre-processing by FM significantly reduces the dissimilarity with the ground truth. See *supplemental material* for more results.

Acknowledgements: The authors wish to thank the Qualcomm Innovation Fellowship Program (India) for supporting this work, NVIDIA Corp. for donation of a Titan Xp GPU, and Prof. Manasa Ranjan Behera of the Civil Engineering Department at IITB, for the wave-flume facility to acquire real data.

References

[1] Github Repository. https://github.com/jeringeo/FourierFlows. 1

[2] Project Page. https://www.cse.iitb.ac.in/~ajitvr/publications.html. 1

[3] M. Alterman, Y. Schechner, P. Perona, and J. Shamir. Detecting motion through dynamic refraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):245–251, 2013. 1

[4] H. Altwaijry, A. Veit, and S. Belongie. Learning to detect and match keypoints with deep architectures. In *British Machine Vision Conference (BMVC)*, 2016. 3

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 3

[6] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005. 4

[7] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011. 7

[8] C. Cox and W. Munk. Slopes of the sea surface deduced from photographs of sun glitter. *Bulletin of the Scripps Inst. Oceanogr.*, 6:401479, 1956. 3

[9] A. Donate and E. Ribeiro. Improved reconstruction of images distorted by water waves. In *Advances in Computer Graphics and Computer Vision*, 2007. 1

[10] A. Efros, V. Isler, J. Shi, and M. Visontai. Seeing through water. In *NIPS*, pages 393–400, 2004. 1

[11] K. Halder, M. Paul, M. Tahtali, S. Anavatti, and M. Murshed. Correction of geometrically distorted underwater images using shift map analysis. *J. Opt. Soc. Am. A*, 34(4):666–673, Apr 2017. 1, 7

[12] S. Henrion, C. W. Spoor, R. P. M. Pieters, U. K. Muller, and J. L. van Leeuwen. Refraction corrected calibration for aquatic locomotion research: application of snells law improves spatial accuracy. *Bioinspiration and Biomimetics*, 10(4), 2015. 1

[13] J. G. James, P. Agrawal, and A. Rajwade. Restoration of non-rigidly distorted underwater images using a combination of compressive sensing and local polynomial image representations. In *ICCV*, 2019. 1, 3, 5, 6, 7

[14] G. Katzir and N. Intrator. Striking of underwater prey by a reef heron, egretta gularis schistacea. *J. Computational Physics A*, 160:517–523, 1987. 1

[15] Z. Li, Z. Murez, D. Kriegman, R. Ramamoorthi, and M. Chandraker. Learning to see through turbulent water. In *WACV*, pages 512–520, 2018. 1, 7

[16] H. Murase. Surface shape reconstruction of a nonrigid transport object using refraction and motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(10):1045–1052, 1992. 1, 2, 3

[17] O. Oreifej, G. Shu, T. Pace, and M. Shah. A two-stage reconstruction approach for seeing through water. In *CVPR*, pages 1153–1160, 2011. 1, 2, 7, 8

[18] Z.-M. Qian and Y. Q. Chen. Feature point based 3d tracking of multiple fish from multi-view images. *PLOS ONE*, 12(6):1–18, 2017. 1

[19] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 5

[20] K. Seemakurthy and A. N. Rajagopalan. Deskewing of underwater images. *IEEE Trans. Image Processing*, 24:1046–1059, 2015. 1, 2, 7

[21] Y. Tian and S. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *ICCV*, pages 2303–2310, 2009. 1, 2, 6, 7, 8

[22] D. G. Turlaev and L. S. Dolin. On observing underwater objects through a wavy water surface: A new algorithm for image correction and laboratory experiment. *Izvestiya Atmosph. Ocean. Phys.*, 49(3):339345, 2013. 1

[23] Z. Wen, D. Fraser, and A. Lambert. Bicoherence: a new lucky region technique in anisoplanatic image restoration. *Appl. Opt.*, 48(32):6111–6119, 2009. 1

[24] Z. Wen, A. Lambert, D. Fraser, and H. Li. Bispectral analysis and recovery of images distorted by a moving water surface. *Appl. Opt.*, 49(33):6376–6384, 2010. 1, 5