Seeing through Water: Image Restoration using Model-based Tracking

Yuandong Tian and Srinivasa G. Narasimhan
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

A video sequence of an underwater scene taken from above the water surface suffers from severe distortions due to water fluctuations. In this paper, we simultaneously estimate the shape of the water surface and recover the planar underwater scene without using any calibration patterns, image priors, multiple viewpoints or active illumination. The key idea is to build a compact spatial distortion model of the water surface using the wave equation. Based on this model, we present a novel tracking technique that is designed specifically for water surfaces and addresses two unique challenges — the absence of an object model or template and the presence of complex appearance changes in the scene due to water fluctuation. We show the effectiveness of our approach on both simulated and real scenes, with text and texture.

1. Introduction

In many imaging scenarios, the camera and the scene of interest are immersed in different media with an interface in-between. Fig. 1 shows a common example where the camera observes the floor of a pool through the water surface. The task is to recover the image of the floor that is severely distorted by water fluctuation. Similar scenarios occur in turbulence imaging[16], astronomy and satellite imaging[15], underwater imaging[10], measuring objects in liquid[23] and tunable liquid lensing[11].

As illustrated in Fig. 1, light transport across the interface is caused by reflection and refraction[1]. This leads to a variety of effects such as non-single viewpoint distortion[21], double images[4], scene distortion[7, 5, 6, 22] and illumination caustics[17] in a single acquired image. Such effects impair the common goal of imaging systems: to record a faithful image of the scene. When the interface is flat and fixed over time, a static light transport model can be developed to eliminate some effects. For example, reducing light reflections by polarization filtering[14] and removing image distortions by using a multi-viewpoint model of the camera[21].

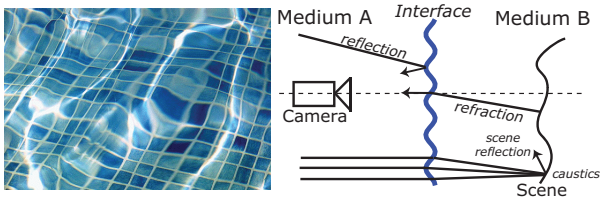

Figure 1. An image captured through a wavy water surface. The camera and the scene of interest are immersed in different media with an interface in between. Complex reflection, refraction effects and caustics can make image interpretation hard. Our technique exploits a physically based model for water surface fluctuation to undistort scenes from a video sequence without requiring calibration objects or active illumination.

On the other hand, when the water surface fluctuates over time, the dependency of light transport on the interface renders any static model inapplicable. In this case, undistortion is possible if the exact shape of the interface can be measured. In [9], the interface and the scene are illuminated by spectrally isolated red and green channels, and captured by a color camera. The surface shape is thus measured by one channel and used to undistort the other. In [12], a known scene (calibration pattern) is tracked to estimate the surface.

When the interface is unknown, a single image is insufficient to recover the scene, and hence a video sequence is necessary. Here, a statistical model is often used to undistort a stationary scene without recovering the water surface. Simple pixel-wise mean/median works well for reducing small fluctuations[18]. A better approach is to select only "good" image patches from the video frames and stitch them together (also known as the "lucky image"[8]). Several works [7, 5, 6] find the center of the distribution of patches from the video as the undistorted patch, either by embedding them on a manifold[7] or by clustering them[5, 6]. Recently, Wen *et al*. [22] model the frames as random phase perturbations in the Fourier domain, and average them using the bispectrum to undistort the image.

In this paper, we address the problem of simultaneously recovering both the water surface and the underlying planar static scene given only a video sequence with severe distortions. The key idea is to build a spatial distortion model

[1] Diffusion into scattering medium like marble/milk is not considered.

of the water surface using the wave equation. In particular, we build a simulator for image distortion due to fluctuations of the water surface (Fig. 2). Since the fluctuations are mostly smooth, we derive a reduced space model of the image distortions using PCA, yielding fewer number of parameters. This enables our model-based tracking algorithm to estimate the water surface at every time instant, by fitting the model to each video frame. Finally, the estimated water surface is used to recover the original scene with reduced distortions. We validate our approach using both simulated scenes and real scenes with text and textures.

Compared to previous approaches, our method has several advantages. We do not require calibration patterns[12], special illumination[9], multiple view points[13], or impose any prior on the scene. We use a short sequence (61 instead of 800 in [7] or 120 in [22]). The water surface estimation enables better stitching of patches than [7] and better stabilization (restoration) of each video frame. In contrast to most tracking methods[3, 24, 2, 1], our approach does not require object models (or templates) and works even with topological changes in the scene. As a trade-off, the model-based tracking requires the acquisition of videos at a higher-than-usual frame rate (125 fps). In the future, we will extend our approach to handle reflections, caustics and higher-order optical effects (double images, total internal reflection, etc).

2. Physical model for water fluctuation

2.1. Image formation

Consider a stationary and planar scene $I_g(\mathbf{x})$ settled at the bottom of the water pool, and an orthographic camera above the water surface, taking images downward. Due to the fluctuating water surface, each video frame $I(\mathbf{x}, t)$ is a distorted version of $I_g(\mathbf{x})$ with the following relationship:

$$I(\mathbf{x}, t) = I_g(\mathbf{x} + \mathbf{w}(\mathbf{x}, t)) \tag{1}$$

where $\mathbf{w}(\mathbf{x}, t)$ is the unknown distortion that varies over time. This model is illustrated in Fig. 2. The goal is to recover both the stationary image $I_g(\mathbf{x})$ and the water surface distortion $\mathbf{w}(\mathbf{x}, t)$ using only the captured video $I(\mathbf{x}, t)$.

2.2. Wave equation

According to Snell's law, under first-order approximation[19], we can relate the distortion function (warping) $\mathbf{w}(\mathbf{x}, t)$ to the height $h(\mathbf{x}, t)$ of water surface at time t:

$$\mathbf{w}(\mathbf{x}, t) = \alpha \nabla h(\mathbf{x}, t) \tag{2}$$

where α is a constant related to water height h_0 when the water surface is still and relative refraction index between air and water.

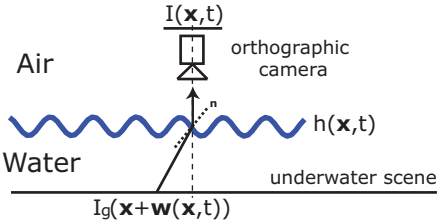

Figure 2. Scene radiance I_g from stationary scene immersed in water is warped due to light refraction at the fluctuating water surface before imaged by a sensor. This relates the distortion function $\mathbf{w}(\mathbf{x}, t)$ to the height $h(\mathbf{x}, t)$ of the water surface at time t.

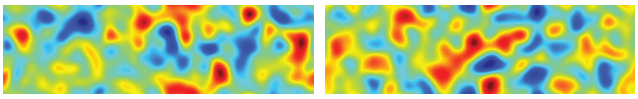

Figure 3. Two samples of 2-D Gaussian processes used as the initial conditions of the wave simulator.

When the maximum surface fluctuation $\max_{\mathbf{x},t} |h(\mathbf{x}, t) - h_0|$ is small compared to h_0, the water surface is governed by the wave equation:

$$\frac{\partial^2 h(\mathbf{x}, t)}{\partial t^2} = c^2 \nabla^2 h(\mathbf{x}, t) \tag{3}$$

where $c = \sqrt{g h_0}$ is the speed of wave (g is the gravity).

Using water equation, we implicitly incorporate all the specific appearance changes induced by water fluctuation into one framework, which eliminates modeling individual water-induced distortions.

2.3. Synthesis of water surface using the wave equation

Given the image formation model, we build a water simulator based on Eqn. 3. Our simulator is more general than that of [22] as we directly simulate the wave equation rather than only use particular solutions of that equation.

To simulate the wave equation, we use forward Euler method which is easy to implement and stable for small time step Δt: $h(\mathbf{x}, t + \Delta t) = 2h(\mathbf{x}, t) - h(\mathbf{x}, t - \Delta t) + c^2 \nabla^2 h(\mathbf{x}, t)(\Delta t)^2$, where $\nabla^2 h(\mathbf{x}, t)$ is the Laplacian operator. The initial conditions $h(\mathbf{x}, 0)$ and $h(\mathbf{x}, 1)$ are chosen to be a spatially correlated Gaussian random processes in a 2-D grid, as illustrated in Fig. 3. More specifically, $h(\mathbf{x}, 0)$ and $h(\mathbf{x}, 1)$ are sampled from a multivariate Gaussian distribution $N(h_0, \Sigma)$ with each entry of the covariance $\Sigma_{\mathbf{x},\mathbf{x}'}$ inversely proportional to the spatial distance between $\mathbf{x}$ and $\mathbf{x}'$:

$$\Sigma_{\mathbf{x},\mathbf{x}'} = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2_{\text{synthesis}}}\right) \tag{4}$$

$\sigma_{\text{synthesis}}$ is set by visually comparing the appearance of a known scene in the water tank with that from simulations. Importantly, $\sigma_{\text{synthesis}}$ is independent of the scene.

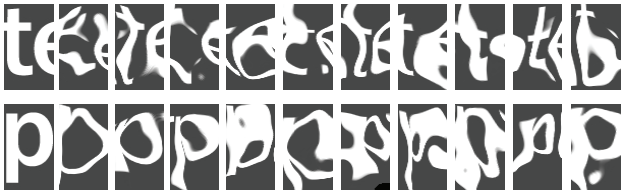

Figure 4. Image synthesis using image formation model (Eqn. 1) and wave equation (Eqn. 2 and Eqn. 3). In each row, the first patch is undistorted; while the rest are distorted.

The motivation for this spatial correlation is that the water surface fluctuates smoothly. Then given the water state $h(\mathbf{x}, t)$, we warp the original image using Eqn. 1 by linear interpolation. Fig. 4 shows the typical distortions observed in synthesized images. Note the topological changes in the letters (splitting or merging parts) and the smooth curvy distortions. All of these appearance changes make it difficult for traditional template/feature tracking methods to work.

3. A reduced model for water distortion

For each video frame of size m_1 by m_2, the water height function $h(\mathbf{x}, t)$ has spatial dimension $m_1 m_2$, which is typically on the order of thousands. Due to the issues of local maxima, time complexity and stability, most parameter estimation methods fail on such a high dimensional space.

Fortunately, since the water surface is smooth, the structure of the height field $h(\mathbf{x}, t)$ can be captured using a small number of dimensions, using model reduction techniques. This step benefits the water shape estimation (Section 4) in two ways. First it reduces the number of parameters so that the estimation process is both computationally feasible and stable. Second, it encodes the smoothness and wave-like characteristics of $h(\mathbf{x}, t)$ into the lower dimensional bases and it is not necessary to explicitly enforce these characteristics (e.g., using smoothness constraints in the objective function) during parameter estimation.

Since it is nontrivial to know the water shape in real data, we use the simulator built in Section 2 to get the training samples, where $c = 0.8\text{pixel}/\text{frame}$ and $\sigma_{\text{synthesis}} = 10\text{pixel}$. We partition the warping $\mathbf{w}(\mathbf{x}, t)$ from simulation into small patches of the same size (57×40), stack them together as training samples, regardless of their exact spatial locations, and compute the PCA bases for these patches. Note $\mathbf{w}$ is a two-dimensional vector field, so the dimension of warping samples is $57 \times 40 \times 2$. In all our experiments, we have used the bases corresponding to the first 10 eigenvalues. The resulting bases $B(\mathbf{x}) = [b_1(\mathbf{x}), b_2(\mathbf{x}), \ldots, b_{10}(\mathbf{x})]$, called *water bases*, are shown in Fig. 5. By construction, the bases are translation invariant.

Once we obtain the bases, we can represent any warping $\mathbf{w}$ of this particular size (57×40) by coefficients $\mathbf{p}$ defined on these bases, i.e., $\mathbf{w}(\mathbf{x}, t) \approx B(\mathbf{x})\mathbf{p}_t$.

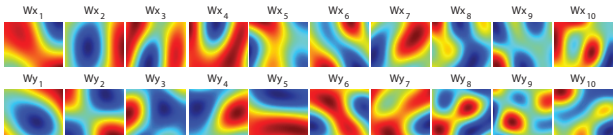

Figure 5. Water bases B.

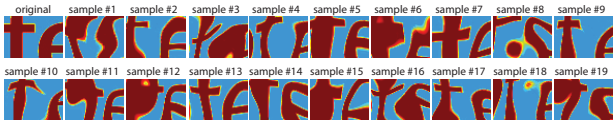

Figure 6. Some samples generated by water bases. Perceptually, these samples look like real images taken under water fluctuation.

One may also use other parametric models, such as splines and polynomials, to obtain the same compact set of bases with similar wave-like behavior. However, many non-intuitive parameters such as coefficients, orders, control points need to be fine-tuned. In contrast, using wave equation achieves the goal intuitively with only 2 parameters (the wave speed c and smoothness $\sigma_{\text{synthesis}}$). Moreover, it is not necessary to precisely determine the value of $\sigma_{\text{synthesis}}$ since the water bases in Fig. 5 offers an additional range of variation in smoothness.

Using patches offers the best trade-off between model flexibility and computational efficiency. Pixel-wise deformation field offers higher degree of freedom, yet it leads to intensive computations and issue of local maxima. On the other hand, model the distortion globally is inflexible and hence fails to capture fine structures.

3.1. Validation

The reduction of the dimensions from $57 \times 40 \times 2 = 4560$ to 10 may seem drastic. There are two reasons for this reduction. First, the exact mean-square errors in reconstructing the simulated data is of less importance to us. What is more important is that the bases capture water wave-like behavior. Second, as we shall discuss in Section 4, more parameters may introduce instabilities in estimation. In the following, we validate our choice of these 10 bases, showing they indeed offer a faithful representation of water distortion.

3.1.1 Samples generated by water bases

We first validate water bases B by generating distorted samples. Fig. 6 illustrates some generated samples. They are perceptually similar to real water-distorted image, despite the low dimensionality of water bases B. The main characteristics of water distortion, such as line distortion and topological changes, are exhibited.

Figure 7. Warp reconstruction using water bases. The first row shows the original image patch, the second row shows the distorted version under water warping simulated using Eqn. 2 and Eqn. 3. The last row shows the distortion obtained by projecting onto the water bases. The similarity between the second and third rows shows the effectiveness of the learned bases.

3.1.2 Warping reconstruction using bases

In addition to generating perceptually plausible samples, the model also has to provide a faithful representation of water distortion. We verify this by comparing original distortions $\mathbf{w}$ and projected distortions $\mathbf{w}_{\text{proj}} = BB^T\mathbf{w}$ on several image patches. The original distortions are sampled from the simulator described in Section 2.

As illustrated in Fig. 7, there is little qualitative difference between a image patch which is distorted by $\mathbf{w}$, and that distorted by projected distortion $\mathbf{w}_{\text{proj}}$ on B.

4. Model-based tracking without template

For each frame of the input video sequence, we must estimate the coefficients $\{\mathbf{p}_t\}$ corresponding to the water bases B. But how do we do this without knowing the underlying undistorted image?

One approach is to exploit temporal continuity of video and develop a tracking algorithm. However, this application presents several challenges for classical techniques that rely on object templates or image features. We develop a new algorithm for tracking water distortions and demonstrate how the coefficients $\{\mathbf{p}_t\}$ can be estimated reliably.

4.1. Challenges for tracking water surfaces

While there has been a lot of progress on tracking a variety of objects[3, 24, 2, 1, 20], a fluctuating water surface poses unique and significant challenges. For the methods where only metric and affine transforms are considered, certain geometric invariants[20] can be used to locate landmarks to help in tracking. However, the distortions due to a wavy water surface are highly non-linear and it is unlikely that simple geometric invariants can be derived.

Several tracking methods also require an object model or a template that can be used in a training stage[3, 24, 2]. However, in our case, no single frame is distortion-free to be used as a template. One option is to use the pixel-wise mean (median) frame as the template. But the mean image is often highly blurred when compared to the original frames.

Alternatively, one may think of using one frame of the input video as the object template and track the rest using the

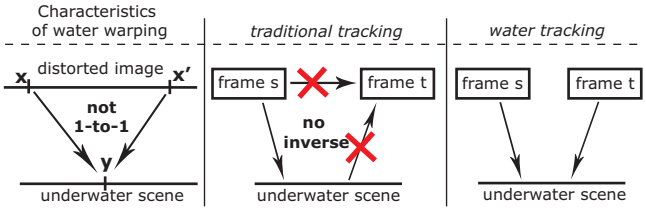

Figure 8. The left image shows water warping maps two distinct positions $\mathbf{x}$ and $\mathbf{x}'$ onto an identical position $\mathbf{y}$ of underwater scene. Water warping is hence not one-to-one and not invertible. Therefore, a tracking paradigm that models the mapping from one frame to the other (shown in the middle) are ill-suited to water warping. Instead, our water tracking technique uses the paradigm on the right, in which no inversion is needed.

warps defined on bases B, as in template tracking[1]. However, the distortion function of a water surface is not one-to-one, hence not invertible (Fig. 8). This means a warping between two images distorted from an undistorted one using B cannot be represented within this subspace. Therefore, traditional tracking techniques are ill-suited.

4.2. Our method

Our goal is to find a consistent underwater image $\hat{I}_g$ such that every frame in the video can be warped to this image using its warping coefficients. Consider two frames $I(\mathbf{x}^s, s)$ and $I(\mathbf{x}^t, t)$ at time s and t as shown in Fig. 8. Using the warping coefficients $\mathbf{p}_s$ and $\mathbf{p}_t$, we can obtain the undistorted image estimates $\hat{I}_g(\mathbf{y}^s, s)$ and $\hat{I}_g(\mathbf{y}^t, t)$ by the following transformation:

$$\mathbf{y}^s(\mathbf{p}_s) = \mathbf{x}^s + \mathbf{w}(\mathbf{x}^s, s) \approx \mathbf{x}^s + B(\mathbf{x}^s)\mathbf{p}_s \quad (5)$$

$$\mathbf{y}^t(\mathbf{p}_t) = \mathbf{x}^t + \mathbf{w}(\mathbf{x}^t, t) \approx \mathbf{x}^t + B(\mathbf{x}^t)\mathbf{p}_t \quad (6)$$

where, $\mathbf{y}^s$ and $\mathbf{y}^t$ are the locations on the two undistorted images respectively. From the image formation model,

$$\hat{I}_g(\mathbf{y}^s, s) = I(\mathbf{x}^s, s), \quad \hat{I}_g(\mathbf{y}^t, t) = I(\mathbf{x}^t, t) \quad (7)$$

Then, our objective is to minimize the difference between the two undistorted images:

$$J(\mathbf{p}_s, \mathbf{p}_t) = \int \left(\hat{I}_g(\mathbf{y}, s) - \hat{I}_g(\mathbf{y}, t)\right)^2 \mathrm{d}\mathbf{y} \quad (8)$$

In practice, we optimize the above objective function in the discrete domain by minimizing the following summation over sample points $\{\mathbf{y}_i^s\}$ and $\{\mathbf{y}_i^t\}$:

$$\tilde{J}(\mathbf{p}_s, \mathbf{p}_t) = \sum_{i=1}^{n} \left(\hat{I}_g(\mathbf{y}_i^s, s) - \hat{I}_g(\mathbf{y}_i^s, t)\right)^2 + \text{symm}. \quad (9)$$

Here, "symm" is the other term with superscript t instead of s, $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ are sampled from regular image (pixel) grid

locations and the coordinates $\mathbf{y}_i^s$ and $\mathbf{y}_i^t$ are obtained using Eqn. 5 and Eqn. 6. From Eqn. 7, we rewrite Eqn. 9 as:

$$\tilde{J}(\mathbf{p}_s, \mathbf{p}_t) = \sum_{i=1}^{n} \left(I(\mathbf{x}_i^s, s) - \hat{I}_g(\mathbf{y}_i^s, t) \right)^2 + \text{symm.} \quad (10)$$

We estimate $\hat{I}_g(\mathbf{y}_i^s, t)$ and $\hat{I}_g(\mathbf{y}_i^t, s)$ by interpolation. For $\hat{I}_g(\mathbf{y}_i^s, t)$, we find nearby sample points $\{\mathbf{y}_j^t\}$ from frame t and use corresponding intensity values $\{\hat{I}_g(\mathbf{y}_j^t, t)\} = \{I(\mathbf{x}_j^t, t)\}$ to interpolate at position $\mathbf{y}_i^s$ and obtain $\hat{I}_g(\mathbf{y}_i^s, t)$. Similarly we obtain $\hat{I}_g(\mathbf{y}_i^t, s)$. To make $\tilde{J}$ smooth, the interpolation is done using kernel regression. In particular, higher weights are given on close neighbors and lower weights on faraway ones. We also interpolate $\hat{I}_g(\mathbf{y}_i^t, s)$.

Eqn. 10 is thus minimized using gradient descent with proper initial conditions to be discussed in Section 4.2.2.

4.2.1 Two heuristics to handle local minima

Minimizing the objective function in Eqn. 10 is a non-convex optimization problem and may result in local minima. We handle this problem using two heuristics. First, we start with optimizing Eqn. 10 over neighboring pairs of frames, as in traditional tracking. Optimizing over frames separated by a large time interval may encounter more local minima, and is only practical when reasonable initial conditions are available. Second, instead of directly working with the solution $(\mathbf{p}_s, \mathbf{p}_t)$ that minimizes Eqn. 10, we regard their difference $\Delta\mathbf{p}_{ts} = \mathbf{p}_t - \mathbf{p}_s$ as the reliable estimation. This is used to avoid any translational ambiguity.

It is tempting to directly use the wave equation to model the continuity between frames. However, three factors make it hard: first, the wave equation cannot be solved without knowing possibly arbitrary external excitation and/or damping forces; second, using wave equation again involves temporal coupling of distortion parameters; finally, since we adopt a patch-based method, modeling the interactions between patches remains nontrivial. In practice, the heuristics presented above work well as shown by our results.

4.2.2 Warping estimation for multiple frames

Eqn. 10 is defined for two frames, while typically a video has many frames. One approach is to extend the idea of estimating a consistent undistorted image over multiple frames by defining a similar function $\tilde{J}(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T)$. However, this function involves joint optimization of all $\{\mathbf{p}_t\}$, which is computationally difficult due to issues of local minima, stability and computational complexity.

Here we adopt two incremental approaches: one for long sequences (61 frames in our experiment) used in image restoration; the other for short sequences (15 frames in our experiment) used in video stabilization.

Long video sequences: To estimate $\{\mathbf{p}_t\}$, we optimize Eqn. 10 over the frame pairs in a particular order. Specifically, given a sequence of frames, we first find the center frame $I(\mathbf{x}, c)$, and estimate the warping difference between $I(\mathbf{x}, c)$ and its neighboring frames $I(\mathbf{x}, c \pm 1)$ using the initial condition $\mathbf{p}_c = \mathbf{p}_{c\pm1} = 0$. Then we optimize Eqn. 10 between the center frame $I(\mathbf{x}, c)$ and $I(\mathbf{x}, c \pm 2)$ using the solution $(\mathbf{p}_c, \mathbf{p}_{c\pm1})$ of previous optimization as the initial condition, and so on.

The reason for this strategy is that neighboring frames are often similar. However, just using neighboring frames independently to obtain $\Delta\mathbf{p}_{tc}$ may lead to accumulated errors or drift. To maintain stable estimation, it is also important to have long-range connections (e.g. between $I(\mathbf{x}, c)$ and $I(\mathbf{x}, 1)$), which is possible with our strategy.

After all the warping differences $\Delta\mathbf{p}_{tc}$ between frame t and center frame c are obtained, we exploit the periodic nature of water fluctuation and estimate $\mathbf{p}_t$ by enforcing the constraint $\sum_t \mathbf{p}_t = 0$ over all the frames.

Short video sequences: In this situation, we just compute the neighboring warping difference $\Delta\mathbf{p}_{t+1,t}$ using Eqn. 10 under initial condition $\mathbf{p}_t = \mathbf{p}_{t+1} = 0$. Then we minimize the following function to obtain coefficients $\{\mathbf{p}_t\}$:

$$R(\mathbf{p}_1, \ldots, \mathbf{p}_T) = \sum_{t=1}^{T-1} \|\mathbf{p}_{t+1} - \mathbf{p}_t - \Delta\mathbf{p}_{t+1,t}\|^2 + \lambda \sum_{t=1}^{T} \|\mathbf{p}_t\|^2 \quad (11)$$

where T is the number of frames. The second term is yet another way to express the periodic nature of water fluctuation because in such a short video, $\sum_t \mathbf{p}_t = 0$ may not necessarily hold. We choose $\lambda = 0.1$ in our experiments.

As a convex optimization problem, Eqn. 11 can be solved efficiently, resulting in a faster way to do video stabilization than the previous strategy. In fact, one can first compute the neighboring warping differences over the entire video sequence, and then optimize Eqn. 11 on each sliding window to obtain a stabilized frame.

4.3. Validation

We verify the tracking procedure on simulated scenes, in which we know the state of water surface and undistorted scene. Samples of the 61 consecutive patches of size 57×40 are shown in Fig. 9. We estimate the coefficients $\mathbf{p}$ on water bases B using the above procedure. Then the undistorted patch is estimated by first undistorting each frame using the estimated warping and computing the mean patch over all 61 frames. Our method results in better image restoration for severe distortions as shown in Fig. 9. The blurring is due to the averaging of individual undistorted frames. We believe this is a really challenging data set and our method produces an image with similar topology while pixel-based methods do not. Also, notice that the estimation of $\mathbf{p}$ are reasonable compared to ground truth.

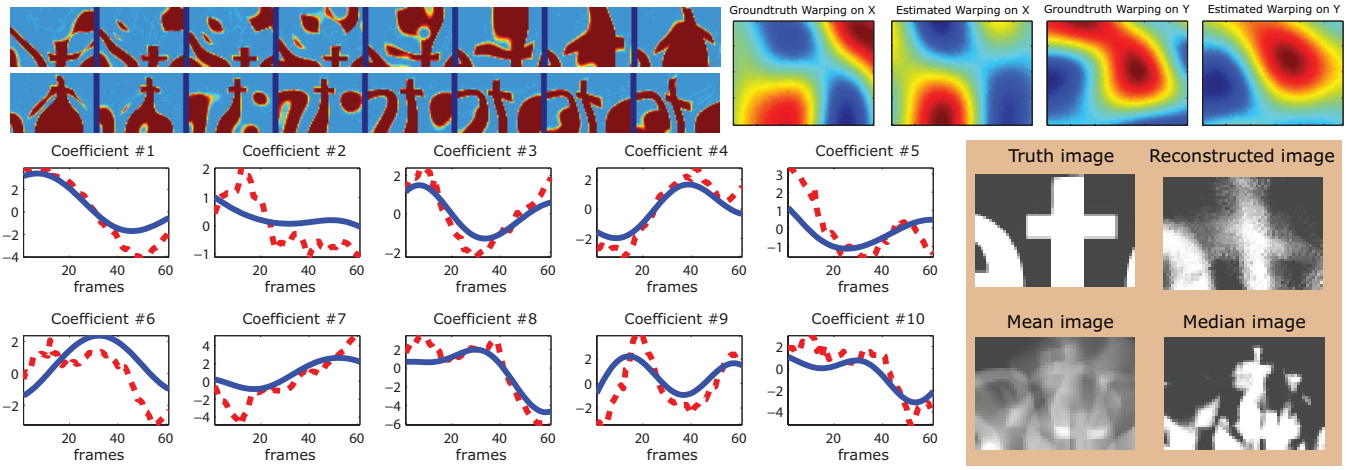

Figure 9. Water tracking validation using simulated data. The two rows on the top-left show sample patch inputs. High distortions including translation, curving and topological changes (merging and splitting) are depicted compared to the original image (shown in bottom-right). Handling these distortions is out of the scope of traditional tracking methods, while the proposed water tracking techniques can. To verify the accuracy of our estimation, the bottom-left part shows estimated coefficients (red dashed line) versus true coefficients (blue solid line) on the water bases B, while the top-right part shows the reconstructed warping in X and Y direction versus the true warping at frame 31 (the middle frame). The estimations, although not perfect, are reasonable given the complex appearance changes.

5. Image restoration and video stabilization

5.1. Experiment setup

The experiment consists of a 125 Hz video camera observing vertically downward a 0.5m deep semi-transparent water tank with a planar scene at the bottom. The tank is illuminated from the side to avoid surface reflections. The water surface is manually disturbed.

5.2. Image restoration

For a patch sequence of the same size as water bases B, we can estimate warping coefficients using the tracking method, and restore the underwater patch. For a video sequence with larger spatial dimension, we first partition the video into disjoint patches of size 57×40, for each patch perform the restoration and stitch the results together. However, it creates artifacts on the patch boundaries, as in [7].

Given that we know the image distortion of each frame tiled by patch distortions $\mathbf{w} \approx B\mathbf{p}_t$, a simple Gaussian blurring on warping boundaries and restoration using these blurred deformation fields alleviates the problem. In contrast, it would be hard if only the restored patches are known.

Fig. 10 shows image restoration results on several scenes with different sizes of text and texture. The first column shows one sample of the 61 input frames, severely distorted by water fluctuation. The second and third columns show the results from pixel-wise approaches. The last two list our results with two different partitions, showing significant improvements especially in the case of text.

We verify the accuracy of the estimated warps as follows.

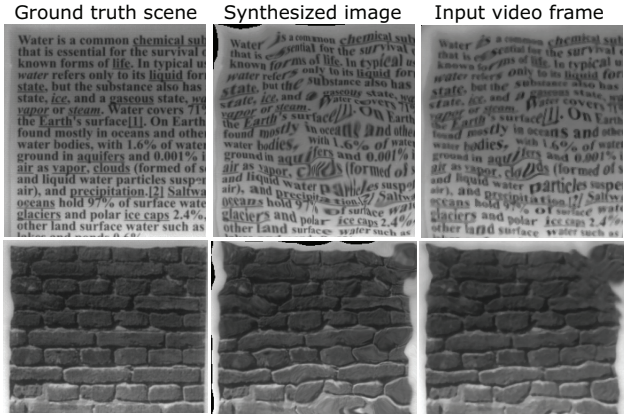

Figure 11. Image synthesis using estimated warping. Given the ground truth scene with still water surface (the first column), a distorted image (the second column) can be synthesized that is similar to corresponding input video frame (the third column). Note the distortion and topological changes are correctly synthesized.

Using the ground truth undistorted scene, we synthesize the distortions using the estimated warps, as shown in Fig. 11. The visual similarity between the synthesized and the acquired frames validates our warp estimation approach.

5.3. Video stabilization

Similarly, we can stabilize a long video[2] by applying our method on continuous temporal windows, as described in Section 4.2.2. We use a sliding window of size 15 (corresponding to 0.12 second at 125 fps). Pixel-wise

[2] Videos can be downloaded at http://www.cs.cmu.edu/∼yuandong

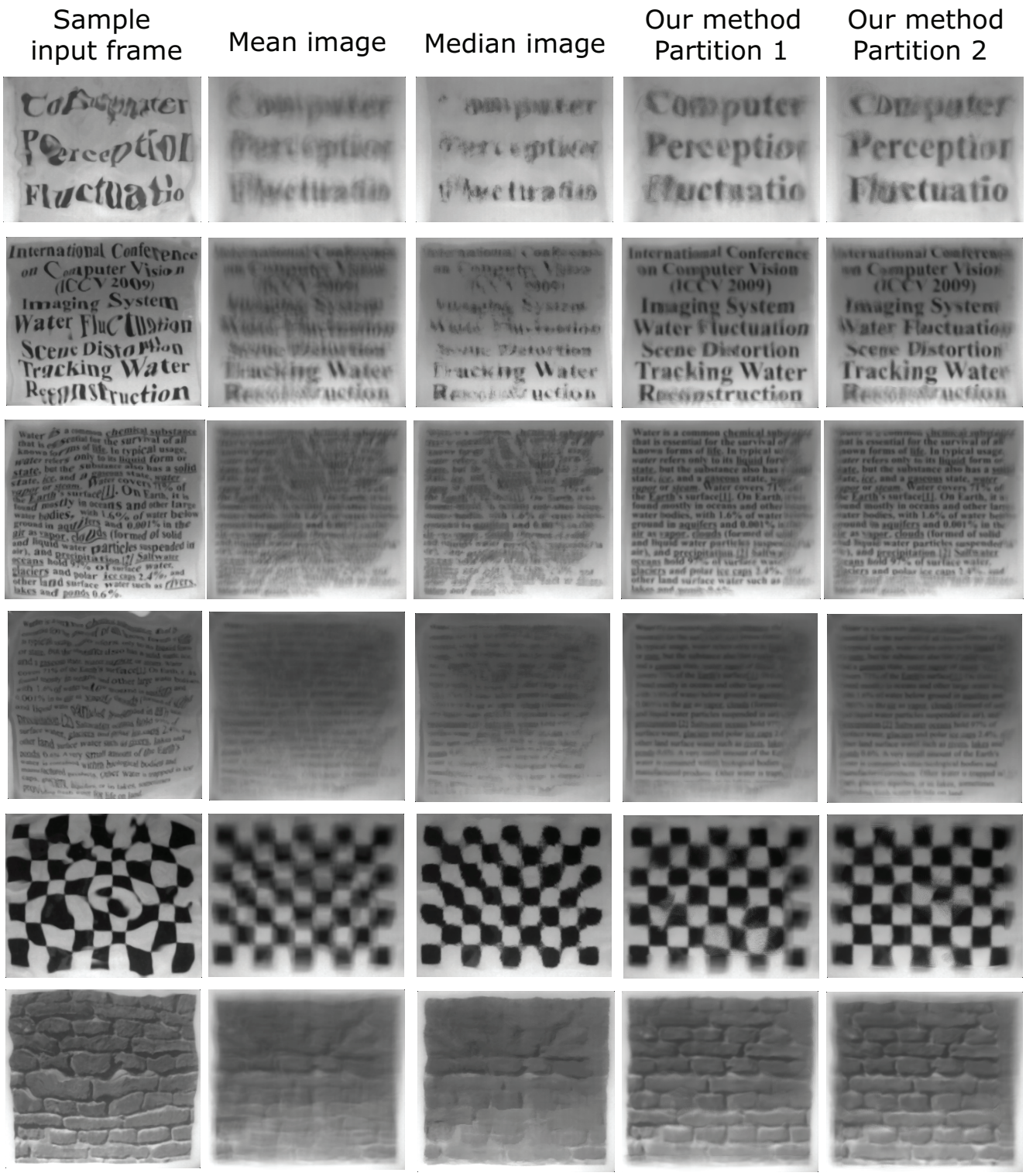

Figure 10. Image restoration results. We test our method on four different sizes of text fonts, as well as check board and brick textures. The first column shows a sample frame from the input video, which is severely distorted. Then the second and third column shows the result by pixel-wise mean/median, which is less distorted but severely blurred. Finally our results are shown on the last two columns using two patch partitions (partition A starts at image coordinate $(1, 1)$, while partition B starts at image coordinate $(28, 20)$, the center of the left-top rectangle in partition A). Notice our method alleviates the distortion but still retains image details (Please zoom in to see the details).

mean/median on the sliding window of the same size are also shown. Compared to the input video and results from mean/median, it is clear that our method reduces water fluctuation with minimal loss of image quality.

6. Discussion

There are many further directions beyond this work. The requirement of a camera with a higher-than-usual frame rate (125 fps) is closely related to the issue of stability and local minima in tracking. It is possible to employ a hierarchical structure on tracking to avoid local minima and hence extend our work to video frames with lower sample rate.

Another avenue is to explore the optimal number and size of water bases. With too few bases, the major characteristics of water warping cannot be described accurately; with too many bases, the tracking procedure becomes slow and unstable, and may encounter more local minima. Similarly there exists a trade-off for the size of bases. On one hand, larger bases cover more image area and thus help stabilize the tracking procedure; on the other hand, more bases have to be involved to compensate for larger size.

In Section 4.3, we compute the pixel-wise mean values of undistorted images to recover the underwater scene, which results in blurring. In the future, we will explore more sophisticated statistical approaches[7, 22].

Undistorting scenes that have undergone significant refraction is a hard problem. Our work provides an initial step in a new direction with potential implications in other domains such as turbulence imaging and liquid lensing.

Acknowledgements: This work was supported in parts by ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762 and an NSF CAREER Award IIS-0643628.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[3] T. Cootes, G. Edwards, C. Taylor, et al. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[4] Y. Diamant and Y. Schechner. Overcoming visual reverberations. In *IEEE Computer Vision and Pattern Recognition*, 2008.

[5] A. Donate, G. Dahme, and E. Ribeiro. Classification of Textures Distorted by WaterWaves. In *Proceedings of the 18th International Conference on Pattern Recognition-Volume 02*, pages 421–424. IEEE Computer Society Washington, DC, USA, 2006.

[6] A. Donate and E. Ribeiro. Improved Reconstruction of Images Distorted by Water Waves. In *International Conference on Computer Vision Theory and Applications*, pages 228–235. Springer, 2006.

[7] A. A. Efros, V. Isler, J. Shi, and M. Visontai. Seeing through water. In *Neural Information Processing Systems (NIPS 17)*, 2004.

[8] D. Fried. Probability of getting a lucky short-exposure image through turbulence. *Optical Society of America, Journal*, 68:1651–1658, 1978.

[9] O. Ju, V. Savchenko, I. Levin, et al. Correction of image distorted by wavy water surface: laboratory experiment. In *Proceedings of the IV International Conference Current Problems in Optics of Natural Waters(ON W'2007), N. Novgorod*, pages 91–93, 2007.

[10] D. Kocak, F. Dalgleish, F. Caimi, and Y. Schechner. A focus on recent developments and trends in underwater imaging. *Marine Technology Society Journal*, 42(1):52, 2008.

[11] S. Kuiper and B. Hendriks. Variable-focus liquid lens for miniature cameras. *Applied Physics Letters*, 85:1128, 2004.

[12] N. Morris. Image-based water surface reconstruction with refractive stereo. Master's thesis, University of Toronto, 2004.

[13] N. Morris and K. Kutulakos. Dynamic refraction stereo. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 2, 2005.

[14] E. Namer and Y. Schechner. Advanced visibility improvement based on polarization filtered images. *Proc. SPIE 5888: Polarization Science and Remote Sensing II*, pages 36–45, 2005.

[15] F. Roddier. The effects of atmospheric turbulence in optical astronomy. *Progress in optics.*, 19:281–376, 1981.

[16] M. Roggemann and B. Welsh. *Imaging through turbulence*. CRC press, 1996.

[17] Y. Schechner and N. Karpel. Attenuating natural flicker patterns. *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, 3, 2004.

[18] R. Shefer, M. Malhi, and A. Shenhar. Waves distortion correction using cross correlation, 2001.

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

[20] T. Tommasini, A. Fusiello, V. Roberto, and E. Trucco. Robust feature tracking in underwater video sequences. In *In Proceedings IEEE Oceans*, pages 46–50, 1998.

[21] T. Treibitz, Y. Schechner, and H. Singh. Flat refractive geometry. In *Proc. IEEE CVPR*, volume 2, 2008.

[22] Z. Wen, D. Fraser, A. Lambert, and H. Li. Reconstruction of Underwater Image by Bispectrum. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, volume 3, 2007.

[23] A. Yamashita, E. Hayashimoto, T. Kaneko, and Y. Kawata. 3-D measurement of objects in a cylindrical glass water tank with a laser range finder. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings*, volume 2, 2003.

[24] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4), 2006.