

Water Surface Modeling from A Single Viewpoint Video

Chuan Li, David Pickup, Thomas Saunders, Darren Cosker, David Marshall, Peter Hall
and Phillip Willis

Abstract—We introduce a video based approach for producing water surface models. Recent advances in this field output high quality results but require dedicated capturing devices and only work in limited conditions. In contrast, our method achieves a good trade off between the visual quality and the production cost: it automatically produces a visually plausible animation using a single viewpoint video as the input. Our approach is based on two discoveries: first, shape from shading (SFS) is adequate to capture the appearance and dynamic behaviour of the example water; second, shallow water model can be used to estimate a velocity field that produces complex surface dynamics. We will provide qualitative evaluation of our method and demonstrate its good performance across a wide range of scenes.

Index Terms—video based modeling, water surface modeling

1 INTRODUCTION

Water simulation has been widely studied in the Computer Graphics literature for many years. Substantial effort has been made to improve rendering quality using physically-based fluids simulations [1–8]. However in some graphics applications, such as [9, 10], a good balance between the visual plausibility and the production cost is more important than perfect physical accuracy. Another example is the “look development” in visual effects, where a director might want to create a scene that is not necessarily physically accurate, but nonetheless has the right “look”. This inspires us to develop a low cost video based approach to produce animations that capture the “look” and “feel” of the water from real world scenes.

Methods have been proposed for automatic reconstruction of complex objects and scenes from images or videos, for example: faces [11], human bodies [12], hair [13] and trees [14]. Among them water brings unique challenges [15]. Major difficulties include its lack of matchable features for stereo reconstruction; and its complex dynamics, including topological changes, yield extreme difficulties for visual tracking.

In this paper we are interested in capturing water surfaces from nature scenes. Differently, we do not aim for generating models that are perfectly accurate, nor animations that surpass the existing state-of-the-art graphics simulations. Instead we follow the recent

video based approaches and explore the possibility of a lower cost method for producing visually plausible water surface animations, such as the examples in Figure 1. Compared to the existing approaches, our method has the following advantages:

- Simplicity – it works with a single input video recorded by an ordinary camera and generates visually plausible results for outdoor scenes.
- Robustness – it performs consistently well across different types of water.
- Efficiency – it only requires a linear optimization so the computational cost is low.

Our technical contribution is to reproduce interesting water surface dynamics from an input video using a relatively simple physical model. Before giving further details, here we first explain the two important discoveries that our method is based on.

Our first discovery is shape from shading (SFS) is able to capture the appearance and dynamic behaviour of the water from the example video. It is true that water surface properties normally violate the Lambertian condition of SFS. However the geometric inaccuracies on such a free-form surface are less obvious to human perception. Instead the dynamics of the surface, such as how the waves evolve with time, play a more significant role. Although SFS is not a perfect reconstruction of the water surface, we observe that it reproduces an animation has a similar dynamic behavior. We will show output animations of various types of water surfaces throughout this paper.

However, SFS only captures the surface geometry not the velocity. To interact with the environment, we need to fit the surface data with a physical model – one outputs a velocity field that reproduces the interesting dynamics of the SFS surfaces. Our second discovery is the shallow water model [16] can be

• C. Li, D. Pickup, T. Saunders, D. Cosker, P. Hall and P. Willis are with the Department of Computer Science, University of Bath, UK. E-mail: c.li, d.pickup, t.r.saunders, d.p.cosker, p.m.hall, p.j.willis@bath.ac.uk

• David Marshall is with the Cardiff School of Computer Science, Cardiff University, UK. E-mail: Dave.Marshall@cs.cardiff.ac.uk

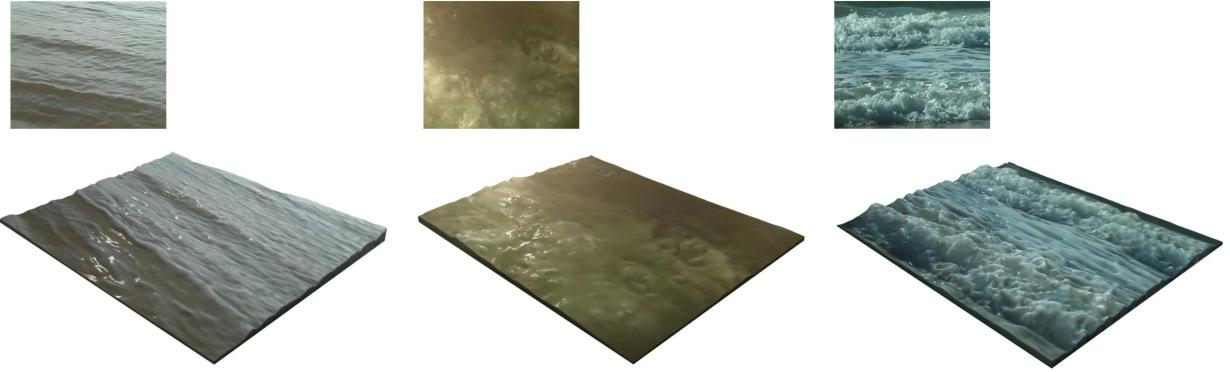


Fig. 1. Examples of our output model for different types of water. Each output (the bottom row) only takes a single video as the input (the top row). The texture from the original video is used to improve the realism of the rendering. Although some scenes, such as the boiling water (second) and the tide (third), violate the shallow water assumption, our system is still able to produce visually plausible models using a multi-layer approach.

used to acquire velocities that faithfully reproduce complex surface dynamics. We conduct experiments across not only a wide range of SFS data collected from the outdoor environment but also 3D scanned data captured in a laboratory condition, and show our method consistently produces good fitness between the model and the data.

The rest of this paper explains our method and evaluates the results.

2 RELATED WORK

A complete model for water surface behavior should contain both geometry and velocity. In this section we review the research of video based water modeling from two related aspects: water surface geometry modeling and fluids tracking.

2.1 Water Surface Geometry Modeling

Video based methods for modeling water surface geometry can be generally divided into two categories: refraction-based methods [17–20] and reflection-based methods [15, 21].

Murase [17] reconstructs a water surface from the apparent motion of an underwater refracted test pattern. The distortion of the pattern is tracked, from which the water's surface normal is calculated using a refraction model. The water surface is then recovered by 2D integration of the surface normal. Balschbach et al. [18] also use a refraction approach, but based on a shape from shading technique where multiple illuminations are used to improve the surface gradients. Morris and Kutulakos [19] show that refractive index is dispensable by assuming light is refracted only once. Their system reconstructs the water surface by minimizing the refractive disparity. These refraction-based methods are generally called “shape from distortion” and they work well for transparent water. The disadvantage is they can not work with less

transparent scenes as specially designed devices are required to be visible underneath the water surface. A slightly different refraction based approach is proposed by Ihrke et al. [20], who dissolved the chemical Fluorescein in the water and measured the thickness of the water from the amplitude of the emitted light. The visual hull of the water surface is then calculated. However, this method requires the Fluorescein to be dissolved in the water so is not practical for large-scale scenes such as rivers and lakes.

Shape from stereo is widely used for reconstructing solid objects from images. With dedicated device setups, it is also used to reconstruct free-form objects such as water. Wang et al. [15] dyed water with white paint and projected light patterns onto its surface. A depth field is first reconstructed by dense reconstruction and then refined using physical constraints. This method shows very accurate reconstructions of surface details but is not adaptable to large scale scenes, such as a lake or a sea. Hilsenstein [21] reconstructed water waves from thermographic image sequences acquired from a pair of infrared cameras and found that infrared stereo reduces the problem associated with transparency, specular reflection and lack of texture at visible wavelengths.

Nonetheless, these techniques all require dedicated experimental setups. Missing from the literature is a low cost solution for capturing water surfaces from a single video captured in an ordinary outdoor environment, as demonstrated in Figure 1. In this case, it is impractical to put special devices under the water, nor dissolve anything in the water. Additionally, the water surface has a lack of recognizable features for shape from stereo techniques to work properly.

2.2 Fluid Tracking

Geometry does not contain the full set of water surface properties. For example, it does not model

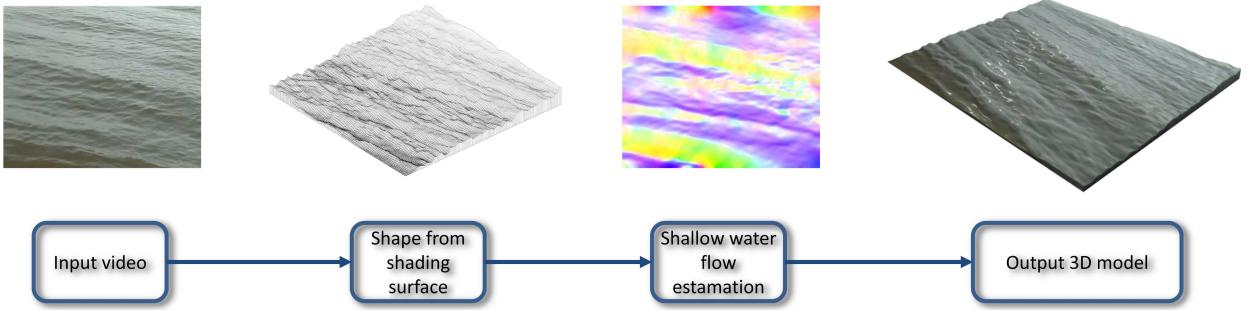


Fig. 2. Our system inputs a single input video and outputs a 3D water surface approximation. It contains two main steps: the first uses shape from shading to acquire an initial water surface, the second estimates a 3D flow field using shallow water as the underlying physics constraints.

the velocity; in particular the horizontal movement is missing. To acquire a flow field from a video, various types of trackers are proposed. Traditional 2D tracking algorithms such as Horn-Shunck optical flow [22] are found to perform less well for water because the conservation of intensity rarely holds. Nakajima et al. [23] propose an energy function as a weighted combination of conservation of intensity, conservation of mass, and momentum equations. Doshi and Bors [24] use a robust kernel which adapts to the local data geometry in the diffusion stage of the Navier-Stokes formulation. Sakaino [25] proposes a method to model abrupt image flow change. Flow is modeled using a number of base waves and their coefficients are found to match the input sequence. Although these methods improve the Horn-Shunck optical flow, their physical constraints are defined in 2D, they are not suitable for 3D acquisition and display.

Pickup et al. [26] is the first attempt to show the simultaneous reconstruction of 3D surface geometry and velocity from a single input video. They combined shape from shading and optical flow to derive a water model that is incompressible in 3D. In this paper we improve on their work in the following aspects: 1) a more complete water model is incorporated to capture the advection field of the flow; 2) a multi-layer framework to further optimize the model is included; 3) qualitative evaluation is conducted to analyse the output model and demonstrate the benefit of our approach.

3 SYSTEM OVERVIEW

Our system inputs a single input video and outputs a 3D water surface approximation. As shown in Fig. 2, it contains two main steps: the first uses shape from shading to acquire the water surface geometry, the second estimates a 3D flow field using SFS data as the geometric constraint and the shallow water equation as the underlying physics model. To cope with complex water movement, a multi-layer framework is used to capture the waves of different scales. The final

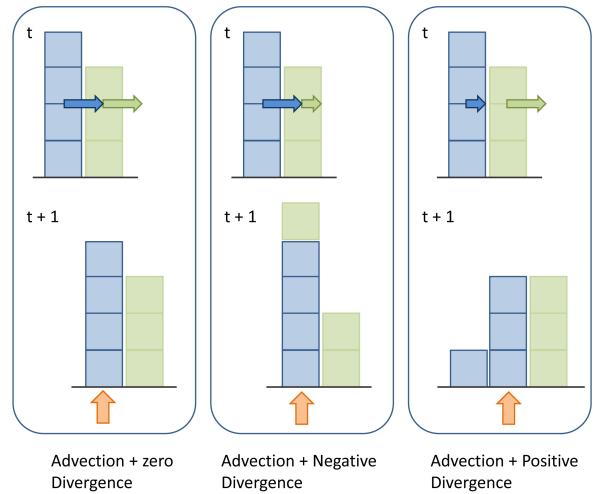


Fig. 3. This figure explains the shallow water model using three examples, each shows the change of a height field at a fixed point (indicated by the red arrow) from time t to $t + 1$. When the horizontal divergence field is zero (the left example, where the blue bar and the green bar have the same velocity), the height field at the fixed point is changed due to the green bar being replaced by the blue bar. Due to the incompressibility of the water, a negative horizontal divergence field will further increase the height field (the middle example) and a positive horizontal divergence field will decrease the height field (the right example).

output water surface (both geometry and velocity) is the superimposition of all layers.

Shallow water is a special case of water simulation that allows fast computation [16, 27]. It represents the water surface as a height field $h(x, y)$, and the water bed as $b(x, y)$. The actual water depth is calculated as $z(x, y) = h(x, y) - b(x, y)$. In this paper we use (u, v, w) to represent the velocities along (x, y, z) directions. Appendix A derives the shallow water model used in this paper for interested readers. The governing

equation is:

$$z_t + z_x u + z_y v + z(u_x + v_y) = 0 \quad (1)$$

This means the evolution of the water surface z_t should be modeled by two terms. The height field is first, advected by the horizontal velocity: $-(z_x u + z_y v)$; and second, increased or decreased proportional to the 2D divergence field: $-z(u_x + v_y)$. Figure 3 gives a graphical illustration of this process.

4 SHALLOW WATER MODEL ACQUISITION

This section introduces our single video based approach to acquire a water model that satisfies the shallow water model property. It is a three-step model fitting process. Firstly, shape from shading is used to acquire an initial surface geometry. Secondly, a physical model is fitted to the shape from shading data. Finally, a multi-layer framework is used to further optimize the fit between the model and the data.

4.1 Data Acquisition

The first step is to acquire some initial data from the input video for model fitting. Figure 4 shows several outdoor water examples with their shape from shading surfaces underneath using Tsai's method [28]. These examples are taken from the the Dyntex Dataset [29]. No complex setting is needed for shooting these scenes: they are recorded by a single static video camera at the speed of 25 frames per second. The resolution of the videos is 352 by 288.

For an input video we process each frame independently. Extreme bright or dark points are removed using low-pass filtering. In practice, SFS surfaces tend to have vertical drifts that are caused by the global luminance changes in the video recording stage. We normalise the average height of each frame to zero by subtracting per-frame means. The bottom row of Figure 5 shows an example of the shape from shading surface successfully evolving with the video.

4.2 Model Fitting

Now we estimate the 3D velocity that is missing from the SFS data. We use SFS surfaces as a prior to constrain the velocity estimation. The velocity is then used in turn to drive the initial SFS surface and produces an animation that conforms to the shallow water equation. Notice we do not intend to improve the SFS surface, which is already a visually plausible approximation of the real water surface. The purpose of the modeling fitting is to acquire a 3D velocity field that enable basic physical interactions.

As indicated by Equation 1, the horizontal advection and divergence fields should satisfy z_t , the Eulerian derivatives of the surfaces with respect to time. Noticing z_t can be directly estimated by subtracting

the two succeeding shape form shading surfaces. Our objective energy function for velocity estimation is a weighted combination of the shallow water equation and a smoothness constraint:

$$E = \iint [(z_t + z_x u + z_y v + z(u_x + v_y))^2 + \alpha^2(|\nabla u|^2 + |\nabla v|^2)] dx dy \quad (2)$$

Notice this is a pairwise estimation and uses surfaces from time t to $t+1$. Here u , v and z are functions for the horizontal velocity fields and the height field. For each time t , they are defined on a grid of the same size as the video frame. Here α is set to unity across all scenes. Equation 2 can be minimized by solving the associated Euler–Lagrange equations. The Euler–Lagrange equations can be expressed into a linear system: $A_u B^T = C_u$, $A_v B^T = C_v$, where $B = [u, v]$ and

$$A_u = [\underbrace{z_x^2 - \frac{\partial}{\partial x}(zz_x) + z^2 + 2\alpha^2}_{a_{u1}}, \underbrace{z_x z_y - \frac{\partial}{\partial x}(zz_y)}_{a_{u2}}] \quad (3)$$

$$A_v = [\underbrace{z_x z_y - \frac{\partial}{\partial y}(zz_x)}_{a_{v1}}, \underbrace{z_y^2 - \frac{\partial}{\partial y}(zz_y) + z^2 + 2\alpha^2}_{a_{v2}}] \quad (4)$$

$$C_u = [\frac{\partial}{\partial x} z^2 u_x + z z_y v_x + (-z z_x + \frac{\partial}{\partial x} z^2) v_y + (z^2 + \alpha^2) \tilde{u} + \alpha^2 \bar{u} + z^2 v_{xy} - z_x z_t + \frac{\partial}{\partial x}(zz_t)] \quad (5)$$

$$C_v = [(-z z_y + \frac{\partial}{\partial y} z^2) u_x + z z_x u_y + \frac{\partial}{\partial y} z^2 v_y + z^2 u_{xy} + \alpha^2 \tilde{v} + (z^2 + \alpha^2) \bar{v} - z_y z_t + \frac{\partial}{\partial y}(zz_t)] \quad (6)$$

See Appendix B for the derivation of the linear system. An iterative scheme is used to compute u and v . For example, the $(k+1)$ th iteration is expressed as follows:

$$\begin{aligned} u^{k+1} &= \frac{1}{a_{u1}a_{v2} - a_{u2}a_{v1}}(a_{v2}C_u^k - a_{u2}C_v^k) \\ v^{k+1} &= \frac{1}{a_{u1}a_{v2} - a_{u2}a_{v1}}(-a_{v1}C_u^k + a_{u1}C_v^k) \end{aligned} \quad (7)$$

Notice $u_x, u_y, v_x, v_y, \tilde{u}, u_{xy}, \bar{u}, \tilde{v}, v_{xy}, \bar{v}$ in C_u and C_v depend on u and v , so they must be updated after each iteration. We use finite differences to calculate these derivatives. For example \tilde{u} and \bar{u} are the average of u along the x and y directions, respectively: $\tilde{u}(x, y) = \frac{u(x+h,y)+u(x-h,y)}{2h}$ and $\bar{u}(x, y) = \frac{u(x,y+h)+u(x,y-h)}{2h}$. The same applies to the derivatives related to v .

Notice our method only uses a linear solver so there is no significant increase of computation cost compared to classical optical flow tracker such as [22].

The water surface geometry can be successively updated from the initial frame using the Eulerian measurement of the vertical velocity: $z_t = -(z_x u + z_y v + z(u_x + v_y))$. These are the final output water surfaces as they and the velocity field conform with

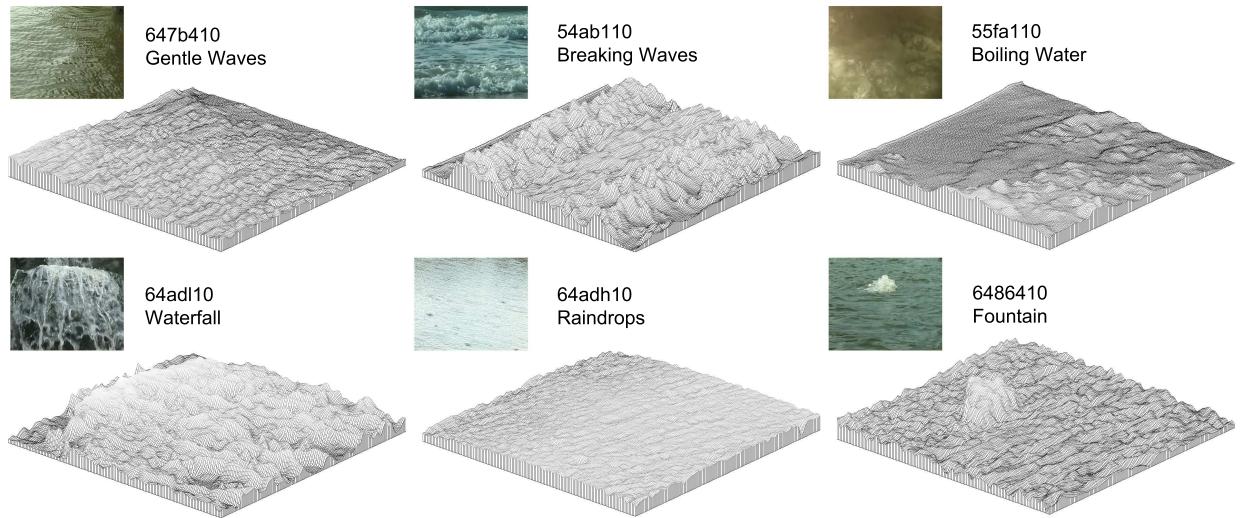


Fig. 4. Some example input videos and their shape from shading surfaces. The surfaces are rendered from a different view point to reveal the 3D information. Here we show six different types of water where each of them represents a unique challenge.

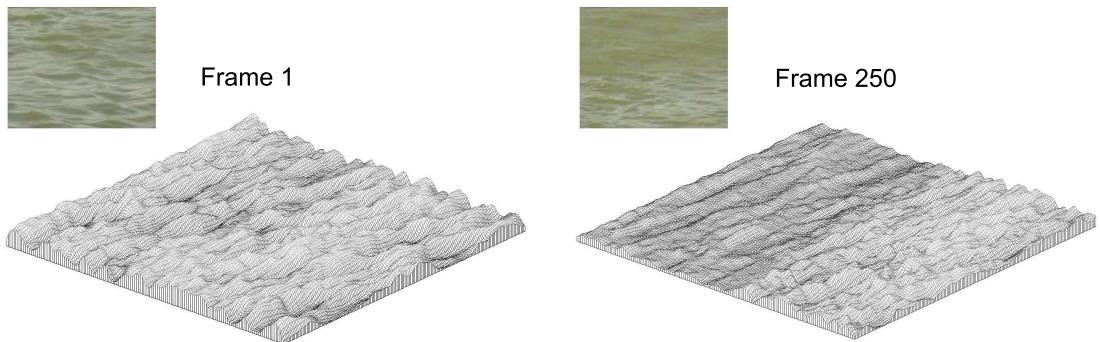


Fig. 5. Here we show the 1st and 250th frames from one water example. Notice the surface successfully evolves with the video.

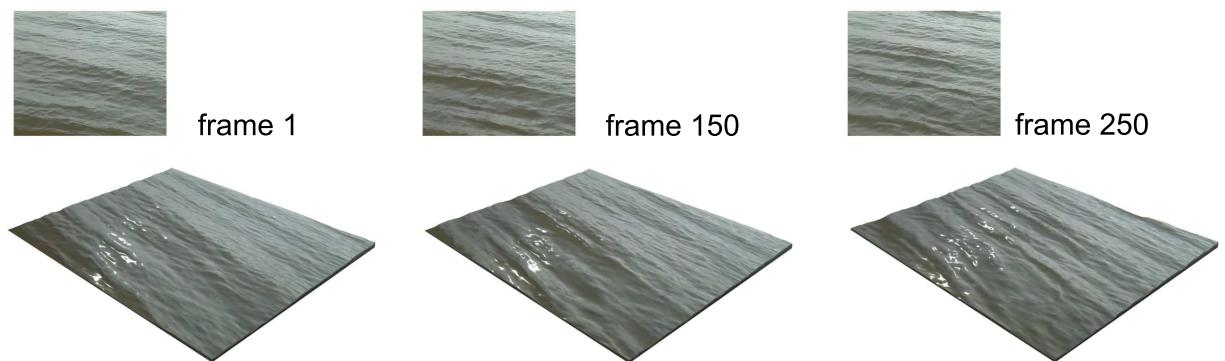


Fig. 6. Our single-layer shallow water model is able to produce animations of gentle water surface movement. This figure shows the output water surfaces (the bottom row) from three different video frames (the top row). They are rendered with textures from the original videos.

the shallow water model. Figure 6 shows three output water surfaces rendered with textures from the original videos.

Notice that the difference between the data (SFS surfaces) and the model (shallow water surfaces) should not be ignored, as indicated by the residual surfaces

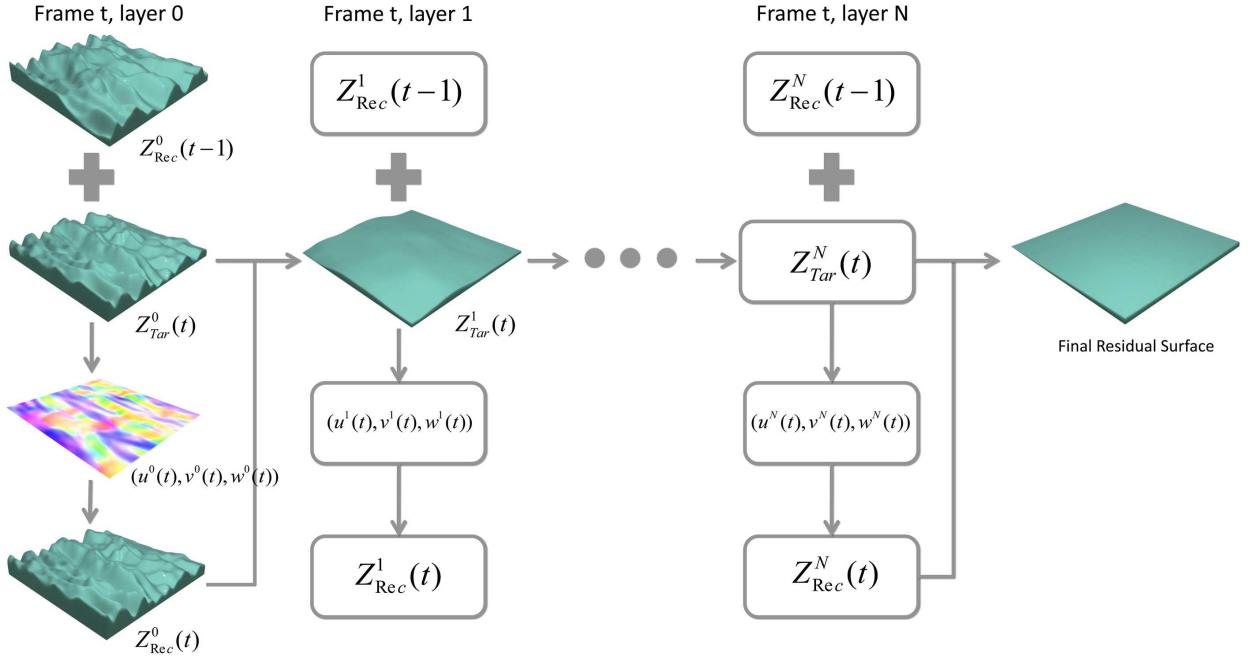


Fig. 7. This figure shows the flowchart of multi-layer shallow water acquisition. The first column shows the process of single layer shallow water model fitting, that is, estimate $(u^0(t), v^0(t), w^0(t))$ and $Z^0_{Rec}(t)$ from $Z^0_{Rec}(t-1)$ and $Z^0_{Tar}(t)$. The residual surface on the current layer is used as the target surface for the next layer, for example, $Z^1_{Tar}(t) = Z^0_{Tar}(t) - Z^0_{Rec}(t)$. The process continues from the left side to the right side of the figure. The difference between the data and the model (the residual surface) is continuously reduced (flattened) until overfitting appears.

in Figure 7. The residual indicates the shallow water equation may not be adequate to describe the full surface dynamics. Without increasing the complexity of the physical model, we propose a multi-layer framework which further reduces the residual surfaces. In this way, the simplicity of the shallow water model is kept and the output model is optimized to better align with the apparent motion in the video.

4.3 Multi-Layer Optimization

Multi-scale based approaches [30] are widely used in applied mathematics. For fluids simulation, multi-scale methods have been proposed to reduce the computational cost while keeping the quality of the rendering high [31]. Our multi-layer shallow water model is acquired from the shape from shading surface with several residual surfaces of different scales. These surfaces form a “target surface pyramid” for each frame. The target surface on layer zero, $Z^0_{Tar}(t)$, is the shape from shading surface. To generate the next layer, we first estimate the 0^{th} layer shallow water velocity $(v^0(t), u^0(t), w^0(t))$. This is then used to recover the 0^{th} layer shallow water surface $Z^0_{Rec}(t)$. The residual between the current data and the model is also calculated: $z^0_{Tar}(t) - z^0_{Rec}(t)$. Notice the residual surface usually has a smooth appearance, indicating a low frequency velocity field that has not been captured on the current layer. To solve this issue, the

residual surface is resized by half and used as the next level target surface, $Z^1_{Tar}(t)$. It is then used to estimate the first layer’s surface velocity and geometry. The rest of the pyramid is generated following a fine to coarse procedure, that is $z^2_{Tar}(t) = z^1_{Tar}(t) - z^1_{Rec}(t)$, $z^3_{Tar}(t) = z^2_{Tar}(t) - z^2_{Rec}(t)$ and so on.

Figure 7 shows the work flow of our multi-layer framework. The gradually flattened target (residual) surfaces indicates the fitting between the model and the video is improved. The final output surface is the sum of all shallow water surfaces: $z(t) = \sum_{i=0}^N z^i_{Rec}(t)$. Here N is the total number of layers. We stop proceeding to the next layer if the residual surface increases, which means an over-fitting. N usually stops at three for our testing videos and the remaining residual is discarded as the artifacts from the data. Figure 8 shows an example of the water surface captured from a tide video sequence using our multi-layer approach.

It is important to notice that the target surface is not a sub-sample of the previous layer’s target surface, but the previous layer’s residual surface. This means the velocity can be estimated independently for each layer; that is, to calculate all frames on level i and then all frames on level $i + 1$. This straightforward multigrid method [30] is different from the pyramid tracking methods, such as the pyramid KLT tracker [32] where the coarse level’s flow is usually used as a prediction for the finer level. The reasons to do

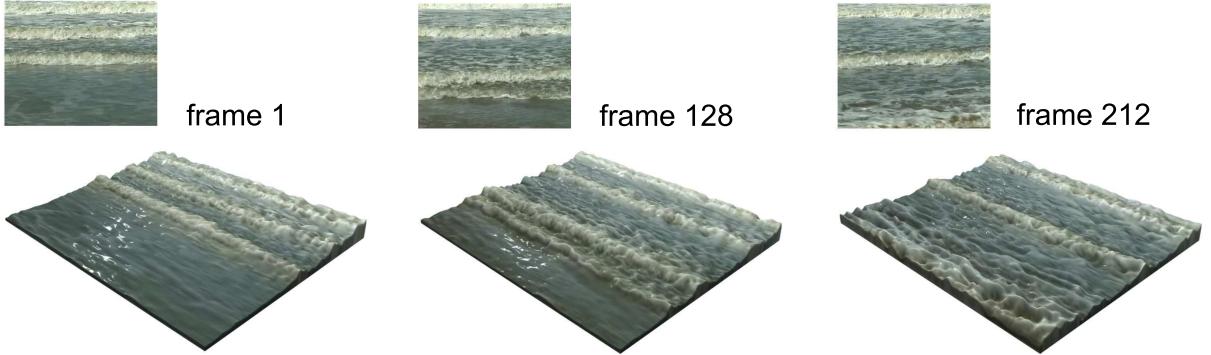


Fig. 8. We use a multi-layer framework to improve the result of complex scenes, such as this tide sequence. Here we show the output surfaces (the bottom row) of three different video frames (top row). Notice our method is able to produce visually plausible results even the underlying physics is not perfectly accurate.

so are : 1) for the advection term to work properly, velocity needs to be small enough so the surface gradient is constant within the magnitude of the velocity (analogous to the constraint of the first order Taylor series approximations in optical flow). Scaling up the velocity and applying it to the finer layer is problematic as the higher resolution surface gradient is not constant within the magnitude of the scaled velocity. 2) For the divergence field to be computed correctly, the velocity should not exceed one pixel. In this way a cell in the water body is forced to receive fluid only from its neighboring cells. Scaling the velocity will break this constraint and lead to numerical instability.

So far we have introduced our modeling process. In summary, the performance of shape from shading on an outdoor video is indeed a pleasing discovery. A 3D velocity field is then estimated using shallow water equations. The multi-layer framework is used to further optimize the fit between the data and the model. The next section will evaluate our results.

5 EVALUATION

Here we explain first our choice of comparators, then the design of the qualitative evaluation and finally the choice of test datasets.

Choice of competitors. We are interested in the surface animation reproduced. We compare our method with Pickup et al.[26], which as far as we know is the latest method that simultaneously captures the surface geometry and velocity from a single viewpoint video. Video based fluids tracking like [23] is also worth comparing with – because it can reproduce an animation by driving the initial surface using the estimated velocity field.

Qualitative evaluation. We aim to show that our method achieves visually plausible approximation of the example water: we provide our results for different types of water, along with their reference input videos for comparison. In order to show the usage

of our 3D velocity field, we provide an example of interacting the water with a synthetic object. Moreover we will show our approach is visually better than recently-published methods.

Data source. For our data source we use a publicly available video dataset and laboratory data of our own. The input videos for SFS surface acquisition are recorded by a single static camera in the outdoor environment. We use the Dyntex Dataset [29] that is publically available. Figure 4 shows some examples of interesting challenges in the dataset. Sequence 647b410 represents water of relatively small, gentle waves. Sequence 54ab110 shows an example of breaking waves, sequence 55fa110 shows boiling water and 64adl10 shows a waterfall. These are examples of more complex dynamics. 64adh10 shows raindrops falling on a pool and 6486410 shows a fountain. These are examples of interaction between different types of water. Besides outdoor footage, we also use 3D scanned surfaces acquired in the laboratory environment. This is to show our model can be fitted to different data sources.

5.1 Qualitative analysis

Here we provide qualitative analysis of our model, using comparisons with alternative approaches. Examples of animated water surfaces are provided in the supplementary video.

Figure 9 visually compares the output models from different methods. Here we show both the velocity and the geometry of the water surface. For velocity, the color code system in [33] is used to render the flow field. It can be seen that Nakajima et al. [23] produces over-smoothed flow fields while Pickup et al. [26] produces over-sharpened flow fields. In contrast, our shallow water velocity appears sharper than [23] and smoother than [26], which indicates a more complete solution is found.

For surface geometry, Figure 9 shows Nakajima et al. [23] do not change the water surface according

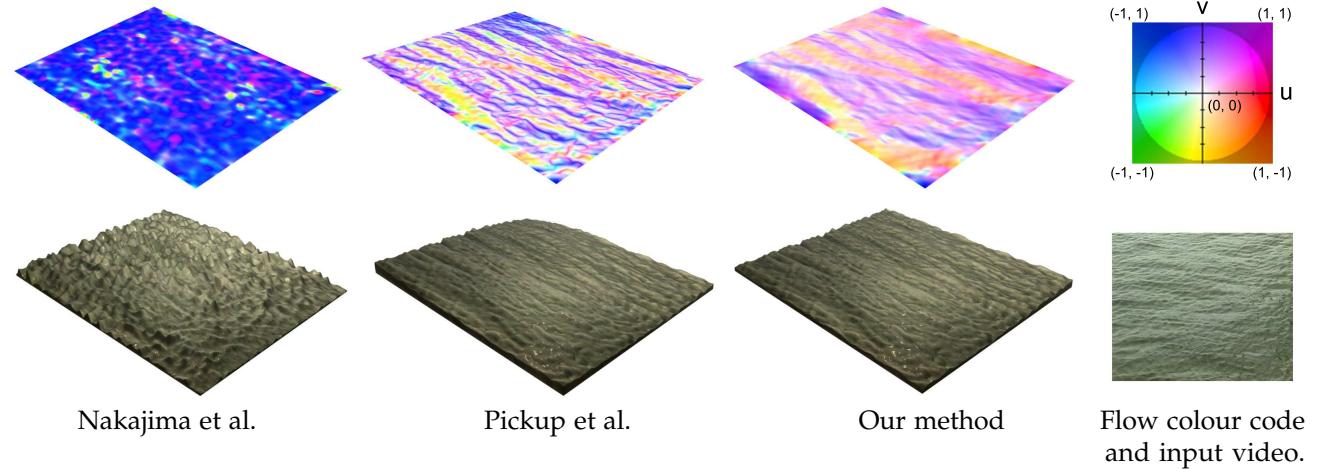


Fig. 9. Here we show a qualitative comparison between different methods. The output velocity is shown on the top and the corresponding output geometry is shown underneath. Notice our output velocity is sharper than Nakajima et al. and smoother than Pickup et al. The reference video frame is also provided for visual comparison with the output surface geometry.

to the input video. They tend to “halt” the water surface due to the lack of vertical movement. As error accumulates in time, the water surface drifts away from its real appearance in the video. The results from Pickup et al. [26] show significant improvement as their water surface continuously evolves with the video. However, there is still a clear difference between the data and the model (indicated by the residual surface error comparison in the supplementary video). Meanwhile there are some common flaws in Pickup et al [26]’s results : 1) large wave features are often poorly modeled (see “comparison: breaking wave example” in the supplementary video); 2) the surface usually drifts away from a horizontal plane as it moves, as shown by the middle column of Figure 9 (also see “comparison: calm wave example” in the supplementary video). The drifting often happens at the boundary of the scene, where water flows into the scene from one side and flows away from another. This artefact is due to the lack of advection in their model: the horizontal divergence field alone just cannot produce enough surface movement to capture the full surface movement. Once there is a large amount of water flow into or out from the video boundary, the divergence field tends to “pump in” or “squeeze out” an extra amount of water. Hence the surface gradually diverts from the horizontal plane. In contrast, the shallow water model benefits from the use of advection field and produces higher quality water surfaces. More examples can be found in Figure 10, 11 and the supplementary video. Examples in Figure 10 and 11 use the input video frames as the water surface texture to create more realistic rendering. The exceptions are the pictures in the bottom row of Figure 10, which are rendered with a new transparent texture and lit with a synthetic environment.

For all examples in Figure 11 we approximate the

water surface as a height field whose horizontal plane is bounded by the video frame, with the vertical axis being opposite to the camera view. It is clear that these examples, especially the waterfall, have physical properties that are very different from shallow water. Nonetheless, our method is still able to produce visually plausible result.

Since our water model contains not only the surface geometry but also the velocity, it is possible to have an object interacting with it. Figure 12 shows an example of a virtual object floating on the water surface. Although a more complete physical simulation can be built upon our model, here we simply ignore the weight of the duck and advect it by the water’s velocity. The body of the object is controlled by a cube, the corners of which are used to sample the velocity vector from the water surface. A 3D similarity transformation (translation plus rotation) is parameterized from these samples and used to transform the body of the virtual box. Users can also scale the transformation to generate an ad hoc simulation of different resistance between the water and the object.

It is also clear that our output model is an imperfect reconstruction due to the limitation of SFS. We show some failure cases in Figure 13. In these cases our output surface is distorted due to the artefact of the shape from shading data. For example, a shadow will generate a valley on the surface. Such an artefact is not recoverable in the shallow water model fitting process. This indicates that better pre-processing is needed to allow our method to work with a wider range of conditions. There is also great potential to make better use of the acquired 3D velocity field. For example, like Thurey et al [27] did, it can be used to drive a layer of particles that will be superimposed on to the shallow water model to produce more interesting features such as breaking waves.

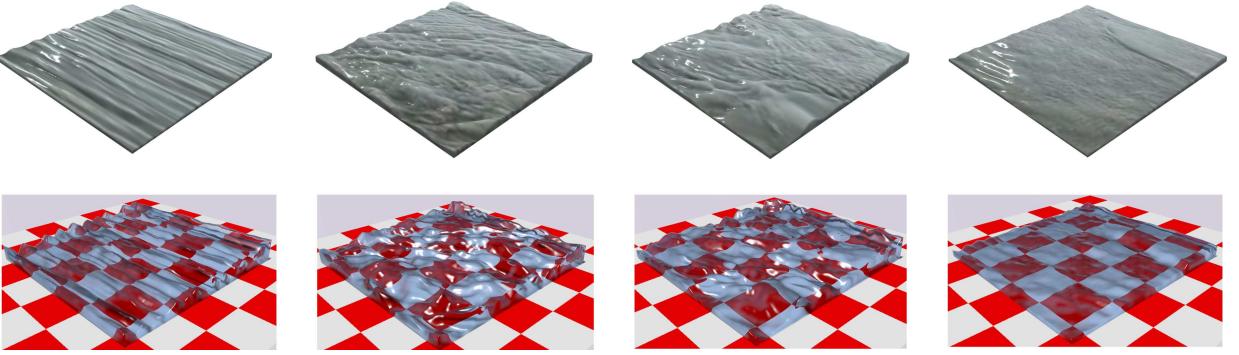


Fig. 10. Here we show four examples of gentle water surfaces captured by our system. The top row shows the results rendered with textures from the input video. The bottom row shows the same surfaces rendered with a transparent material.

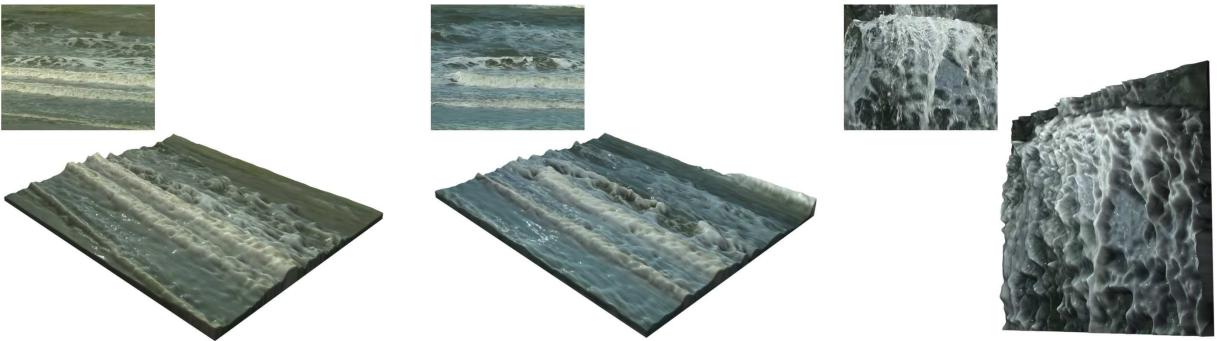


Fig. 11. Here we show three examples of complex water surfaces. Notice although our height field representation can not generate real breaking waves nor foams or splashes, the output surfaces still look plausible with the help of textures from the input video.

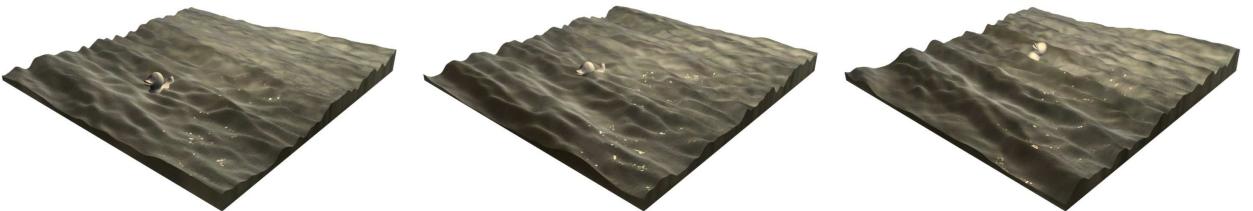


Fig. 12. This figure shows a duck is floating on and advected by the 3D velocity field estimated using our method.

6 CONCLUSION

This paper introduced a low cost solution for acquiring a visually plausible water surface model from a single input video taken in the natural world. Shape from shading is first used to capture the water surface geometry. We then use the shallow water model to estimate the 3D velocity that is missing from the SFS data. In this paper we showed the effectiveness of using such a simple physics model to approximate gentle outdoor waves. To cope with complex water movement, a multi-layer framework is used to capture the waves of different scales. The final output model is the superimposition of all layers. Our method achieves a good trade off between the

visual quality and the production cost. Compared to the existing methods, our method has the following advantages:

- It works fully automatically and only needs a single input video recorded by an ordinary camera and generates visually plausible model. In contrast recent advances in the field of video based fluids modeling require dedicated capturing devices and only work in more limited conditions.
- It uses simple physics assumptions so the model only requires a linear optimization. The computational cost is equivalent to a pyramid implementation of the Horn-Shunck flow tracker [22].
- It performs consistently well across different

types of water. Although some of the scenes violate the SFS and the shallow water assumptions, the results show that our approach is still able to produce visually plausible outputs.

Our technical contributions are based on two interesting discoveries: first, shape from shading (SFS) is able to capture the appearance and dynamic behaviour of the example water; second, shallow water model can be used to estimate a velocity field that produces complex surface dynamics. We have provided qualitative evaluation to support these discoveries.

There are many interesting future extensions of this work. First, our height field representation is not capable of creating fine water details such as splashes, droplets or sprays. One possible solution is to use our output as a base model for highly detailed water simulation. A similar strategy was adopted by [31] for particle simulation. Temporal anti-aliasing has to be investigated for modeling extremely fast moving scenes from low frame rate input videos. Our method is also limited by the artefacts of SFS, which does not handle shadow and reflection of the water surface very well and results in geometric distortion. This indicates a better pre-processing of the video input can be incorporated to improve the results. Last but not the least, our method can be used as a base for data-driven water animation. For example, we are interested in patching together the acquired surfaces in both the spatial and the time domains for synthesizing larger and longer water animations. Although the visual quality of the output model may not be as high as a full physics based simulation, it provides a user friendly approach for accelerating the process of look development in visual effects production.



Fig. 13. Here we show two failure cases of our method. In both cases the surface has serious geometry distortions due to the shadow and the reflection.

7 ACKNOWLEDGEMENTS

We thank reviewers for their valuable suggestions. We thank UK's Engineering and Physical Sciences Research Council for supporting this work with grant EP/D064155/1, and for supporting the Media Technology Research Centre and the Centre for Digital Entertainment at University of Bath. This work is partially supported by the Open Projects Program of NLPB, the Natural Science Basic Research Plan in Shaanxi Province of China (2010JM8005) and the NSFC Grant (61072105).

REFERENCES

- [1] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graphical Models and Image Processing*, pp. 471-483, 1996.
- [2] J. Stam, "Stable fluids," *ACM SIGGRAPH*, pp. 121-128, 1999.
- [3] O.Y. Song, H. Shin and H.S. Ko, "Stable but nondissipative water," *ACM Transaction On Graphics*, pp. 81-97, 2005.
- [4] C. Yuksel, D.H. House and J. Keyser, "Wave particles," *ACM SIGGRAPH*, pp. 99-106, 2007.
- [5] R. Narain, J. Sewall, M. Carlson and M.C. Lin, "Fast animation of turbulence using energy transport and procedural synthesis," *ACM SIGGRAPH Asia*, pp. 1-8, 2008.
- [6] M. Lentine, W. Zheng, and R. Fedkiw, "A novel algorithm for incompressible flow using only a coarse grid projection," *ACM SIGGRAPH*, 2010, pp. 1C9.
- [7] N. Heo and H.-S. Ko, "Detail-preserving fullyeulerian interface tracking framework," *ACM SIGGRAPH Asia*, 2010, pp. 1C8.
- [8] M. Nielsen and R. Bridson, "Guide shapes for high resolution naturalistic liquid simulation," *ACM SIGGRAPH*, pp. 1C8, 2011.
- [9] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bülthoff, "Image-based material editing," *ACM SIGGRAPH*, pp. 654C663, 2006.
- [10] D. Gutierrez, J. Lopez-Moreno, J. Fandos, F. Seron, M. Sanchez, and E. Reinhard, "Depicting procedural caustics in single images," *ACM SIGGRAPH Asia*, pp. 1C9, 2008.
- [11] A. Ghosh, T. Hawkins, P. Peers, S. Frederiksen, and P. Debevec, "Practical modeling and acquisition of layered facial reflectance," *ACM SIGGRAPH Asia*, pp. 1C10, 2008.
- [12] E. D. Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM SIGGRAPH*, vol. 27, no. 3, pp. 1C10, 2008.
- [13] S. Paris, W. Chang, O. I. Kozhushnyan., W. Jarosz, W. Matusik, M. Zwicker, and F. Durand, "Hair photobooth: geometric and photometric acquisition of real hairstyles," *ACM SIGGRAPH*, 2008, pp. 1C9.
- [14] P. Tan, G. Zeng, J. D. Wang, S. B. Kang, and L. Quan, "Image-based tree modeling," *ACM SIGGRAPH*, 2007, pp. 87C93.
- [15] H. M. Wang, M. Liao, Q. Zhang, R. G. Yang, and G. Turk, "Physically guided liquid surface modeling from videos," *ACM SIGGRAPH*, 2009, pp. 1C11.
- [16] R. Bridson, "Fluid simulation for computer animation," A K Peters, 2008.
- [17] H. Murase, "Surface shape reconstruction of a nonrigid transport object using refraction and motion," *IEEE Trans. on Pattern Anal. and Mach.*

- Intell., vol. 14, no. 10, pp. 1045C1052, 1992.
- [18] G. Balschbach, J. Klinke, and B. Jähne, "Multi-channel shape from shading techniques for moving specular surfaces," ECCV, 1998, pp. 170C184.
- [19] N. J. Morris and K. N. Kutulakos, "Dynamic refraction stereo," ICCV, 2005, pp. 1573C1580.
- [20] I. Ihrke, B. Goldluecke, and M. Magnor, "Reconstructing the geometry of flowing water," ICCV, 2005, pp. 1055C1060.
- [21] V. Hilsenstein, "Surface reconstruction of water waves using thermographic stereo imaging," Image and Vision Computing New Zealand, 2005, pp. 102C107.
- [22] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, vol. 17, 1981, pp. 185C203.
- [23] Y. Nakajima, H. Inomata, H. Nogawa, Y. Sato, S. Tamura, K. Okazaki, and S. Torii, "Physics based flow estimation of fluids," Pattern Recognition, vol. 36, no. 5, 2003, pp. 1203C1212.
- [24] A. Doshi and A. G. Bors, "Navier-stokes formulation for modelling turbulent optical flow," BMVC, 2007, pp. 1C10.
- [25] H. Sakaino, "Motion estimation method based on physical properties of waves," CVPR, 2008, pp. 1C8.
- [26] D. Pickup, C. Li, D. Cosker, P. M. Hall, and P. Willis, "Reconstructing mass-conserved water surfaces using shape from shading and optical flow," ACCV, 2010, pp. 189C201.
- [27] N. Thürey, M. Müller-Fischer, S. Schirm, and M. Gross, "Real-time breakingwaves for shallow water simulations," Proceedings of the Pacific Graphics, pp. 39C46, 2007.
- [28] P. Tsai and M. Shah, "Shape from shading using linear approximation," Image and Vision Computing, vol. 12, pp. 487C498, 1994.
- [29] R. Péteri, S. Fazekas, and M. J. Huiskes, "Dyntex: a comprehensive database of dynamic textures," Pattern Recognition Letters, vol. 31, no. 12, 2010, pp. 1627C1632.
- [30] W. L. Briggs, V. E. Henson, and S. F. McCormick, "A multigrid tutorial: second edition," Society for Industrial and Applied Mathematics, 2000.
- [31] S. Barbara and M. Gross, "Two-scale particle simulation," ACM SIGGRAPH, 2011, pp. 1C8.
- [32] J.Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," Intel Corporation Microprocessor Research Labs Report, 2000.
- [33] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," IJCV, vol. 92, no. 1, pp. 1C31, 2011.



Chuan Li Chuan Li is a research officer in the Department of Computer Science, University of Bath. He received his BEng from Zhejiang University in 2005 and his PhD from University of Bath in 2010. His research interests include computer graphics, computer vision and machine learning. He has published research papers in image/video based modeling for natural phenomena.



David Pickup David Pickup graduated from the University of Bristol in 2009 with an MEng in Computer Science. He is currently a PhD student within the CDE and MTRC at the University of Bath. His research focuses on Computer Vision assisted animation of natural phenomena for Computer Graphics.



Thomas Saunders Thomas R. Saunders is a fourth-year PhD student in the Department of Computer Science at the University of Bath. His research interests include sensor-assisted computer vision with applications in camera tracking and dynamic textures. He obtained a M.Eng degree in Electronics and Communications Engineering at the University of Bath in 2009.



Darren Cosker Dr Cosker is currently a Lecturer (UK)/Assistant Prof. (US) at the University of Bath, and also a Royal Society Industry Fellow working with Double Negative Visual Effects (DNeg) applying computer vision and graphics to movies. His current research interests lie in the convergence of Computer Vision, Computer Graphics and Psychology - particularly in relation to human motion analysis and synthesis.



these research areas. He is a member of the IEEE Computer Society and the British Machine Vision Association.



Peter Hall Peter Hall is a reader (associate professor) in Computer Science at the Univ. of Bath, and director of the Media Technology Research Centre. His research interests centre around the applications of computer vision to computer graphics problems, most recently the acquisition of models of complicated natural phenomena using video as a data source.



Phillip Willis Phil Willis is Head of the Department Computer Science at the University of Bath and directs the national research Centre for Digital Entertainment. His research interests are in computer graphics and the representation of images.