3D Shape from Silhouettes in Water for Online Novel-View Synthesis

Tomohiko Yano[1,a)] Shohei Nobuhara[1,b)] Takashi Matsuyama[1,c)]

1. Introduction

This paper is aimed at presenting a new algorithm for full 3D shape reconstruction and online free-viewpoint rendering of objects in water. The applications include (1) education and entertainment such as free-viewpoint 3D visualization of underwater scenes for digital aquariums in future, and (2) 3D analysis of underwater objects and events such as fertilized eggs and their developments.

Suppose we have an object in a water tank of unknown shape. The object is observed by sparsely arranged multi-view cameras via a curved refractive surface of the tank as shown in **Fig. 1**. The goal is to reconstruct the 3D shape of the object from its projections with refractive distortions for online free-viewpoint rendering.

The key contribution of this paper is twofold. We first introduce our calibration model in order to deal with the refractive projection. We then provide a new 3D shape reconstruction algorithm based on shape-from-silhouette (SfS) concept. We also propose an online free-viewpoint rendering system as a practical application.

2. Appearance-based calibration of projections via curved refractive surfaces

The goal of the appearance-based calibration is to obtain a function $f : p \to \ell$ which takes a pixel p in the camera image and returns the corresponding 3D ray line ℓ in the water (**Fig. 2**).

The first design factor is how we parameterize the 3D ray in water. We use two distinctive 3D points, called *near* and *far* points hereafter. In addition, we force each of the near and the far points for all pixels to lie on 3D planes called *near* and *far* clips respectively. This is a crucial point for our 3D shape reconstruction as described later.

The second point is how we provide the training dataset. The key factor is the field coverage. That is, given a training dataset, the learned mapping should be able to cover the entire region of the water tank in the image.

To this end, we employ a dual chess pattern on a plane as shown in **Fig. 3**. The bottom side is in water and captured via the refractive projection. The top side is kept observable

[1] Graduate School of Informatics, Kyoto University
[a)] yano@vision.kuee.kyoto-u.ac.jp
[b)] nob@vision.kuee.kyoto-u.ac.jp
[c)] tm@vision.kuee.kyoto-u.ac.jp

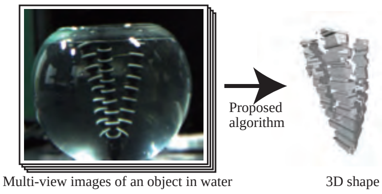

Fig. 1 3D shape reconstruction of objects in water from multi-view images with refractive distortions. Left: a tripod (upside down) in a fishbowl. Right: reconstructed 3D shape.

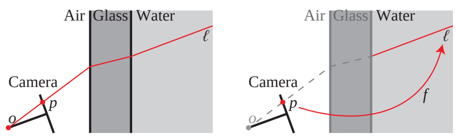

Fig. 2 Appearance-based calibration. The relationship between a pixel p and the ray ℓ which passes through p and the camera center o (left) is modeled by a function $f : p \to \ell$ without knowing the geometry of the refractive layers (right).

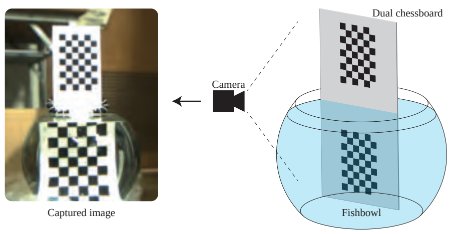

Fig. 3 Dual chess pattern

directly from the camera. This design can limit the feasible size of the tank, but we believe it is a practical solution for regular configurations.

Suppose we can detect and identify the top and bottom chess corners in images $I_i (i = 1, \ldots, n)$ as $p_i^k (k = 1, \ldots, K)$ and $q_i^j (j = 1, \ldots, J)$ respectively. Then the camera pose R_i and t_i w.r.t. the top chess pattern can be calibrated using Zhang's algorithm [4] as $\lambda_i^k p_i^k = A(R_i P^k + t_i)$, where λ_i^k is the projective depth, A is the intrinsic parameter given a priori, and P^k is the model coordinate of the chess corner. Also we can estimate the relative rotation and translation of the two chess coordinates by capturing the board outside the water beforehand. Namely $P = RQ + t$, where P and Q are the model coordinates of the top and bottom chess coordinate systems, and R, t describe their relative rotation and translation.

Up to this point, we could establish a correspondence between a 3D point c_i^j described in the camera coordinate

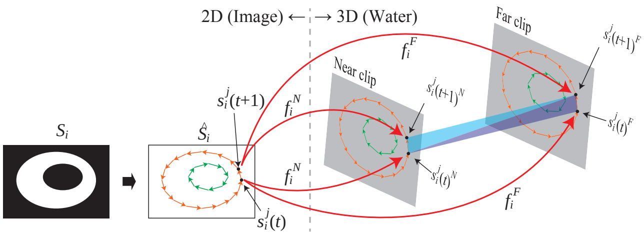

Fig. 4 Visual frustum generation

system and its projection q_i^j in the image I_i. Here points $c_i^j (j = 1, \ldots, J)$ are on a 3D plane in water by definition.

As proposed in [3], learning the mapping to a particular plane in the camera coordinate system can be done by knowing a set of corresponding triplets $\langle q_i^j, Q^j, c_i^j \rangle \, (j = 1, \ldots, J)$. The mapping function f_i should be chosen to reflect the geometry of the refractive surface. For a sphere-like fishbowl (Fig. 3), we employed cubic polynomial functions in practice.

3. Shape-from-silhouette for objects in water for online novel-view synthesis

As is well known, SfS can be implemented in different styles for their own goals. In this paper, we propose a yet new concept *visual-frusta-intersection* designed for our calibration model.

The appearance-based calibration in Section 2 has two important characteristics: (1) asymmetric computation costs for the forward and backward projection computations, and (2) 3D ray space representation by points on parallel virtual planes called near and far clips. To explot these points, we propose the following algorithm.

3.1 Algorithm

Suppose an object in water is captured by N cameras, and produces N-view silhouettes $S_i (i = 1, \ldots, N)$. Given such N-view silhouettes, the goal here is to generate N visual frusta fed to CSG process.

Fig. 4 illustrates the outline of the proposed algorithm. Each of the cameras is supposed to be calibrated beforehand. The calibration parameters of ith camera include the extrinsic parameter R_i, t_i, and the 2D-to-3D mapping functions f_i^N and f_i^F. Here f_i^N and f_i^F return 3D points on the virtual near and far planes respectively.

The first step is to represent the silhouette contour as a set of closed curves $\hat{S}_i$. We employ the non-zero winding rule [1] to identify the holes in order to make the following mesh generation process simpler. That is, curves defined in CCW order fill their interior, while those in CW order cull them out.

The second step is to generate triangles to form the side surface of the visual frusta. Let s_i^j be jth closed curve in $\hat{S}_i$, and s_i^j be discretized into a chain of points $s_i^j(t) \; (t = 1, \ldots, |s_i^j|)$. Back-projecting $s_i^j(t)$ by f_i^N and f_i^F yields corresponding 3D points $s_i^j(t)^N$ and $s_i^j(t)^F$ on the near and far clips respectively. By defining triangles using triplets of points $\langle s_i^j(t)^N, s_i^j(t)^F, s_i^j(t+1)^F \rangle$ and $\langle s_i^j(t)^N, s_i^j(t+1)^F, s_i^j(t+1)^N \rangle$ in this order, thanks to the non-zero winding rule, all the triangles will direct their normals to the outside of the frusta by definition. This is a prerequisite for computing CSG correctly.

The third step is to generate triangles to form the top and bottom planes of the visual frusta. Since the mapping f_i^N and f_i^F are designed to return 3D points lie on planes, $s_i^j(t)^N$ and $s_i^j(t)^F$ form closed polygons on the near and far clips respectively. Therefore, this process is identical to tessellating the polygon on a plane into triangles, and is commonly available as a GPU function.

Finally, by using triangles generated in the 2nd and the 3rd steps, the visual frusta V_i is defined as a triangle mesh which is ready to be fed to Constructive-Solid-Geometry (CSG) [1].

Notice that the proposed algorithm requires small computation costs only: the bitmap silhouette to the contour representation, 2D-to-3D projections using the mapping functions, and the triangle mesh generation. In other words, most of the computation costs on generating the visual hull are offloaded to the CSG part.

4. Evaluation

Fig. 1 shows an example of the captured input images, and 3D shapes given by our method. The processing costs for the visual frusta generation and novel-view rendering were approximately 10ms and 20ms per frame, using Intel Core-i7 3.07GHz CPU and nVidia GeForce GT 640 GPU with an image-based CSG [2]. These values indicate that the proposed scheme is capable of online, real-time novel view synthesis.

5. Conclusion

This paper proposed a 3D shape reconstruction algorithm called *visual frusta intersection* for objects in water. The key idea is to develop an appearance-based calibration which allows obtaining the 3D ray in water corresponding to a pixel in captured images, without knowing the geometry of the refractive layers lying between the camera and the object.

Our future work includes photo-hull reconstruction, texture-mapping, 3D shape estimation of semi-transparent objects, *etc.*

Acknowledgments

This study is partially supported by JSPS Kakenhi 25540068.

References

[1] Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F.: *Computer Graphics: Principles and Practice*, Addison-Wesley, second edition (1996).

[2] Goldfeather, J., Hultquist, J. P. M. and Fuchs, H.: Fast Constructive Solid Geometry Display in the Pixel-Power Graphics System, *Proc. of ACM SIGGRAPH*, pp. 107–116 (1986).

[3] Trifonov, B., Bradley, D. and Heidrich, W.: Tomographic reconstruction of transparent objects, *Proc. of Eurographics Conf. on Rendering Techniques*, pp. 51–60 (2006).

[4] Zhang, Z.: A flexible new technique for camera calibration, *IEEE TPAMI*, Vol. 22, No. 11, pp. 1330–1334 (2000).