Multi-view stereo of an object immersed in a refractive medium

Robin Bruneau, Baptiste Brument, Lilian Calvet, Matthew Cassidy, Jean Mélou, Yvain Quéau, Jean-Denis Durou, François Lauze

Multi-view stereo of an object immersed in a refractive medium

Robin BRUNEAU[a,b,*], Baptiste BRUMENT[a], Lilian CALVET[c], Matthew CASSIDY[a], Jean MÉLOU[a], Yvain QUÉAU[d], Jean-Denis DUROU[a], François LAUZE[b]
[a]IRIT, UMR CNRS 5505, Université de Toulouse, France
[b]DIKU, Department of Computer Science, Copenhagen, Denmark
[c]OR-X, Balgrist Hospital, University of Zurich, Zurich, Switzerland
[d]Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France

Abstract. In this article, we show how to extend the multi-view stereo (MVS) technique when the object to be reconstructed is inside a transparent – but refractive – medium, which causes distortions in the images. We provide a theoretical formulation of the problem accounting for a general, non-planar shape of the refractive interface, and then a discrete solving method. We also present a pipeline to recover precisely the geometry of the refractive interface, considered as a convex polyhedral object. It is based on the extraction of visible polyhedron vertices from silhouette images and matching across a sequence of images acquired under circular camera motion. These contributions are validated by tests on synthetic and real data.

Keywords: 3D reconstruction, multi-view stereo, refraction.

* rb@di.ku.dk

1 Introduction

Natural History Museums often house valuable specimens in transparent mediums, like insects in amber or animals in formaldehyde (see Figure 1). These specimens are crucial for evolutionary studies but challenging to digitise due to the need for 3D see-through techniques. CT scans are a standard method but are costly and not feasible for large collections. Photogrammetric 3D scanning presents a viable alternative, though it faces challenges due to refraction effects and the shape of the interface between air and the medium. We propose a 3D reconstruction method for objects in a homogeneous refractive medium. Building on previous works,[1,2] this paper contributes a revised algorithm for calculating the shortest optical path between a 3D point and its projection in the target image, for interfaces of any shape. Its main contribution, however, is a comprehensive 3D reconstruction pipeline for objects in refractive media.


Figure 1 Left: prehistoric beetle trapped in amber (seen under a microscope). Right: reptiles specimens in jars. Images: A. Solodovnikov (left) and A. D. Jordan (right), courtesy of the Natural History Museum of Denmark.

Assumptions This paper introduces a novel 3D reconstruction method for objects within a refractive medium under the following assumptions: we assume homogeneous refractive media and smooth interfaces, a given index of refraction (or a range), binary masks of the object and of the interface, known camera parameters and triangular mesh representing the interface. In practice, assuming multiple views at fixed rotations on a turntable and visible edges of the medium, camera extrinsics and a convex polyhedron representing the interface can automatically be recovered.

Paper organisation We start by reviewing existing studies on refraction in Section 2. Section 3 details the adaptation of the multi-view stereo technique in the presence of an interface, focusing on predicting the image projection of a 3D point in the refractive medium, a computationally difficult part. Synthetic image tests in Section 4 validate this method. Validation with real data, shown in Section 5, involves developing a robust 3D reconstruction method for polyhedral interfaces and an innovative technique for estimating index of refraction without specialised equipment. The paper concludes in Section 6, suggesting further extensions.

2 Related work

Refraction in computer vision varies in treatment: as a bias to correct in classic vision techniques, an element in active systems, or a feature in refractive 3D reconstruction pipelines.

Refraction compensation in classic vision. Lenses converge light rays from point sources at an *image point*, essential in optical instruments. Precise lens alignment minimises *aberrations*, or undesired refraction effects. Transparent objects in a scene can distort the appearance of opaque objects behind them. Studies have addressed these distortions, particularly with transparent objects like window panes attached to cameras, allowing calibration for standard 3D reconstruction pipelines. Maas in[3] showed how refraction through aquarium glass improves photogrammetry measurements. Łuczyński et al. in[4] corrected images from underwater cameras to restore epipolar geometry. Image pre-correction has been explored in[5,6], with neural network-based correction in[7]. Light field cameras for refraction correction are discussed in[8,9].

Active refraction techniques. Studies termed *active refraction* use refraction for single-view 3D reconstruction, duplicating images using bi-prisms[10,11] or rotating glass plates[12,13].

Estimation of a refractive interface. Morris utilised refracted patterns on water surfaces[14], and with Kutulakos, mapped points seen through transparency[15]. Ben-Ezra and Nayar[16] fit surface models to distorted images of known geometries. Neural network advancements for 3D reconstruction of transparent objects are noted in[17,18].

Bathymetry. Refraction correction is essential in remote-sensing bathymetry, and is exemplified by Murase,[19] Woodget,[20] and Cao.[21]

Classical framework with refraction adaptation. 3D vision systems have adapted to refractive interfaces, covering calibration, camera pose estimation, and techniques like refractive structure-

from-motion, refractive multi-view stereo, and refractive photometric stereo. Sturm[22] discussed camera models for structure-from-motion, including refractive axial cameras. Chari and Sturm[23] extended epipolar geometry for planar interfaces. Łuczyński et al.[4] proposed a pinhole/axial camera model with calibration, and Chen et al.[24] studied fringe projection systems. Challenges in underwater camera use and implications are detailed in works by Jordt et al.[25–27] and others. Pose optimisation under flat refractive interfaces is discussed in[28], with validations primarily on underwater images[29]. Scenarios like viewing aerial objects from underwater are covered in[30,31], and air to water transitions in[32]. Underwater photometric stereo extensions are investigated in studies like[33–36].

Inverse rendering and novel views. Differentiable rasterisers[37–39] and ray tracing inverse renderers[40–42] are emerging in inverse rendering, alongside NeRF adaptations for refraction[43,44]. NeuS[45] and its updated version[46] combine neural SDF (signed distance function) and radiance fields for 3D reconstructions. A framework for objects in cuboid refractive mediums[47] incorporates ambient lighting and ray tracing with Snell-Descartes and Fresnel laws, yet its results are not available for comparison.

We focus on 3D reconstruction by multi-view stereo in refractive media, building upon previous works like patch-based MVS,[48] Kang et al.,[49] and Agrawal,[50] targeting also non-planar interfaces, a gap in current research.

3 From multi-view stereo to refracted multi-view stereo

3.1 Multi-view stereo

Multi-view stereo (MVS) aims to maximise photometric coherence across different images in a 3D scene for dense 3D reconstruction, as summarised in[51]. Given $t+1$ images and their camera

poses, the image of the first pose is chosen as the *reference image*. Let $\mathbf{P}$ denote a 3D point visible in all images, $\mathbf{p} = \pi(\mathbf{P})$ its projection in the reference image and $\mathbf{p}_j = \pi_j(\mathbf{P})$, $j \in \{1, \ldots, t\}$, its projections in the t other images, called *control images*. The Lambertian assumption is written:

$$I_j \circ \underbrace{\pi_j \circ \pi_z^{-1}(\mathbf{p})}_{\mathbf{p}_j} = I(\mathbf{p}), \quad j \in \{1, \ldots, t\} \tag{1}$$

where I_j and I denote the grey level functions of the j-th control and reference images. The index z in π_z^{-1} is necessary as the point $\mathbf{P} = \pi_z^{-1}(\mathbf{p})$ is defined only if its *depth* z is known.

The MVS technique consists in searching for the point $\mathbf{P} = \pi_z^{-1}(\mathbf{p})$, conjugate of $\mathbf{p}$, satisfying the system of Equations (1), by solving, for instance, the least squares problem:

$$\min_{z \in \mathbb{R}} \sum_{j=1}^{t} \left[I_j \circ \pi_j \circ \pi_z^{-1}(\mathbf{p}) - I(\mathbf{p}) \right]^2 \tag{2}$$

In practice, the comparison between the grey levels I_j and I is performed between neighbourhoods of $\mathbf{p}_j$ and of $\mathbf{p}$, the use of a robust estimator is recommended (see the overview presented in[51]).

When the medium is homogeneous, the π_z^{-1} transformation from the reference view to the 3D scene consists in inverting the central projection. Denoting by $\mathbf{K}$ the camera's calibration matrix, this transformation is written:

$$\pi_z^{-1}(\mathbf{p}) = z \ \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \tag{3}$$

The *reprojection* on the j-th control image is also obtained by central projection, considering the camera pose change as a known rigid transformation between the reference pose and the j-th with rotation matrix $\mathbf{R}_j$ and translation vector $\mathbf{t}_j$. With the projection operator $f\left([a, b, c]^\top\right) =$

$[a/c, b/c]^\top$, this second transformation is written:

$$\pi_j(\mathbf{P}) = f\left(\mathbf{K}\left(\mathbf{R}_j\,\mathbf{P} + \mathbf{t}_j\right)\right) \tag{4}$$

Carrying over (3) and (4) into (2), the problem of 3D reconstruction by MVS is rewritten:

$$\min_{z\in\mathbb{R}} \sum_{j=1}^{t} \left[I_j \circ f\left(\mathbf{K}\left(z\ \mathbf{R}_j\ \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} + \mathbf{t}_j\right)\right) - I(\mathbf{p}) \right]^2 \tag{5}$$

The objective in (5) is nonlinear, non-differentiable and/or non-convex, making optimisation potentially difficult. Solving (5) is thus usually done by an exhaustive search (*brute-force*) in a predefined list of values of depth z (see Figure 2-a). This simplistic strategy has shown to be very effective for the 3D reconstruction of scenes with sufficiently textured surfaces[52]. As Figure 2-b indicates, the scenario is more complex when the 3D scene is immersed in a refractive medium.

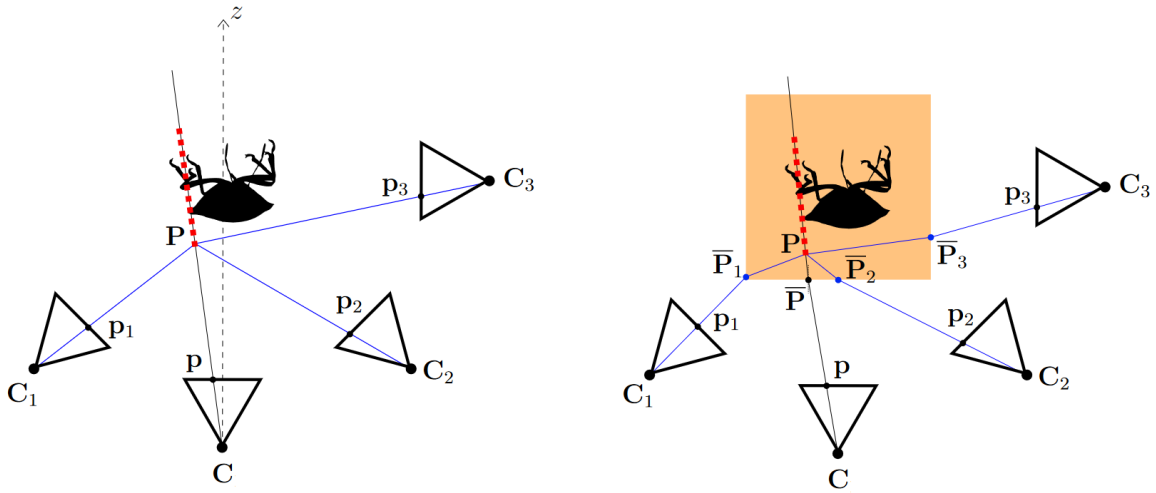


Figure 2 (a) MVS in a homogeneous medium: the different proposals for the point $\mathbf{P}$, which are materialised by red dots, are reprojected in the control images. (b) MVS with refraction: the reprojection of $\mathbf{P}$ in the control images is more difficult to compute, due to refraction.

3.2 Refractive multi-view stereo

The image of an object in a refractive medium, with an *index of refraction* (IoR) over 1, becomes distorted, altering its epipolar geometry. In this context, a point in one image correlates to a curve whose form is influenced by the IoR and the interface shape between the medium and air. Chari and Sturm's work in[23] generalises epipolar geometry's matrix formalism with a 12×12 fundamental matrix, important for camera pose estimation in structure-from-motion. Since refraction adaptation in this field is covered in[27,53], our paper focuses on adapting the MVS technique for 3D scenes in refractive mediums. This new challenge, *refractive multi-view stereo* (RMVS), involves addressing (2) at each point $\mathbf{p}$ in the reference image, with necessary adjustments. In the context of refraction:

- Back-projection of image point $\mathbf{p}$: The back-projection of $\mathbf{p}$ in refractive conditions involves tracing a broken line from $\mathbf{C}$ through $\mathbf{p}$ (see Figure 2-b). The back-projection formula is more complex than (3), expressed as:

$$\pi_{\bar{z}}^{-1}(\mathbf{p}) = \overline{\mathbf{P}} + \bar{z}\,\mathbf{v} \tag{6}$$

  Here, $\overline{\mathbf{P}}$ is the *point of incidence* at the interface, the unit director vector $\mathbf{v}$ of the refracted ray follows Snell-Descartes refraction law (see Section 3.3), and $\bar{z} \geq 0$ is the distance between $\overline{\mathbf{P}}$ and $\mathbf{P}$ along the refracted ray (see Figure 2-b). Determining $\overline{\mathbf{P}}$ varies in complexity with the interface's shape, while computing $\mathbf{v}$ is straightforward if interface normals are accurately known. Tests on synthetic images (with known normals) and real images (assuming a polyhedral interface) are conducted, leaving generalisation to any interface shape and effects of normal estimation inaccuracies for future exploration.

- Reprojection of 3D point $\mathbf{P}$: Computing the reprojection $\mathbf{p}_j = \pi_j(\mathbf{P})$ with refraction is more complex than (4), involving solving a *shortest optical path* problem (see Section 3.3).

- Image multiplication: Refraction can cause a single 3D point $\mathbf{P}$ to project to multiple image points, as shown in Figure 6. Each projection is equally viable for solving (2).

Compared to the MVS technique, the main difficulty of RMVS is the reprojection $\mathbf{p}_j = \pi_j(\mathbf{P})$, $j \in \{1, \ldots, t\}$, of a 3D point $\mathbf{P}$ into the different control images. Let us first consider the case of a planar interface, before tackling the case of an interface of any shape.

3.3 Planar interface

The first Snell-Descartes law asserts that the refracted ray lies in the *plane of incidence*, spanned by the incident ray, and the interface normal in $\overline{\mathbf{P}}_j$: the phenomenon is planar (see Figure 3-a).

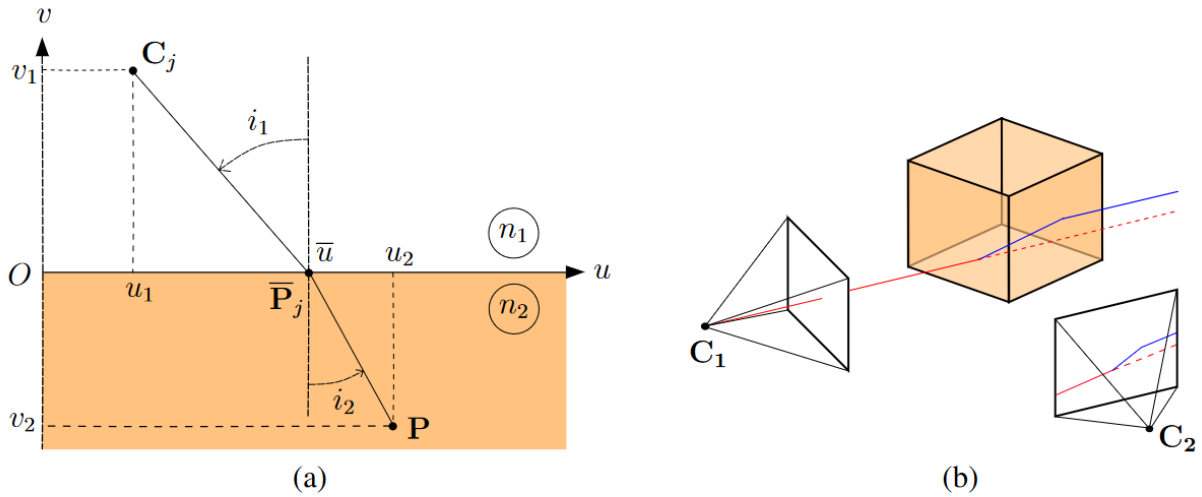


Figure 3 (a) Second Snell-Descartes law on refraction. (b) The back-projected ray, which has two breaks as it crosses the refractive cube, does not project into the control camera along the red epipolar line. For reasons of clarity, this graphical representation does not perfectly conform to the Snell-Descartes laws.

Let i_1 be the angle between the interface normal and the ray in IoR n_1 medium, and i_2 be the angle between the normal and the ray in IoR n_2 medium. The second Snell-Descartes law asserts

that:

$$n_1 \sin i_1 = n_2 \sin i_2 \tag{7}$$

For a planar interface, squaring both sides of (7) and using notations from Figure 3-a, we get:

$$n_1^2 \frac{(u_1 - \overline{u})^2}{(u_1 - \overline{u})^2 + v_1^2} = n_2^2 \frac{(u_2 - \overline{u})^2}{(u_2 - \overline{u})^2 + v_2^2} \tag{8}$$

To find the point of incidence $\overline{\mathbf{P}}_j$ on the u-axis, we need to solve a quartic equation in $\bar{u}$:

$$a_4 \, \overline{u}^4 + a_3 \, \overline{u}^3 + a_2 \, \overline{u}^2 + a_1 \, \overline{u} + a_0 = 0 \tag{9}$$

whose coefficients are based on u_1, v_1, u_2, v_2, and $\alpha = n_2/n_1$[50]. For planar interfaces, (9) typically has one real solution, found using methods like Newton-Raphson. To compute $\mathbf{p}_j = \pi_j(\mathbf{P})$, first solve (9) for $\overline{\mathbf{P}}_j$, then project it into the j-th control image as per (4).

3.4 Interface of any shape

The *Huygens-Fresnel principle* predicts wave surfaces orthogonal to light rays. Dijkstra's algorithm[54] offers a discrete method to calculate these wave surfaces, enabling the shortest path identification between graph vertices. For tracing light rays, the scene can be divided into voxels, serving as the vertices of an undirected graph. The process simplifies in a homogeneous refractive medium, where light propagates straight, similar to air.

The path of a light ray from a 3D point $\mathbf{P}$ to the center of projection $\mathbf{C}_j$ of a control camera, $j \in \{1, \ldots, n\}$, forms a broken line with a single break at the interface, as illustrated in Figure 2-b. As previously discussed, locating the shortest optical path between $\mathbf{P}$ and $\mathbf{C}_j$ boils down to finding

the incidence point $\overline{\mathbf{P}}_j$. Solving this for a planar interface equates to solving a quartic equation (see Section 3.3), but it becomes analytically challenging with more complex interface shapes.

One might wonder if the π_j transformation preserves point alignment, specifically if the π_j image of a refracted light ray remains straight in the j-th control image. Figure 3-b shows that this can be the case with a planar interface, however, with a continuous interface as in Figure 17, the ray image in no more straight.

For general cases, finding the incidence point $\overline{\mathbf{P}}_j$ involves discretising the interface and minimising the optical path of the ray $(\mathbf{C}_j, \widehat{\mathbf{P}}, \mathbf{P})$ through potential points $\widehat{\mathbf{P}}$ on the discretised interface, via a potentially heavy exhaustive search on "eligible points" $\widehat{\mathbf{P}}$:

$$\overline{\mathbf{P}}_j = \underset{\widehat{\mathbf{P}}}{\operatorname{argmin}} \left\{ n_1 \, d(\mathbf{C}_j, \widehat{\mathbf{P}}) + n_2 \, d(\widehat{\mathbf{P}}, \mathbf{P}) \right\} \tag{10}$$

Here, $d(\cdot, \cdot)$ denotes the Euclidean distance in $\mathbb{R}^3$.

Practically, the interface is discretised into a 3D mesh with triangular faces. The eligible points $\widehat{\mathbf{P}}$ for incidence point search are the barycenters of the mesh triangles visible from the projection center $\mathbf{C}_j$, as depicted in Figure 4. The solution of Problem (10) corresponds to the blue-coloured triangle in Figure 5. To refine this result, $\overline{\mathbf{P}}_j$ is then sought in the plane of this triangle, following the method in Section 3.3. The solution is accepted if it is inside the triangle. If not, a similar search is conducted on all adjacent triangles (coloured purple in Figure 5). In the absence of a solution within the triangles, the initial solution of Problem (10) is chosen as the incidence point. A more precise search involving optimisation under linear constraints defining the triangle is possible but significantly increases computation time. Therefore, despite testing this approach, it has been omitted from our current methodology.

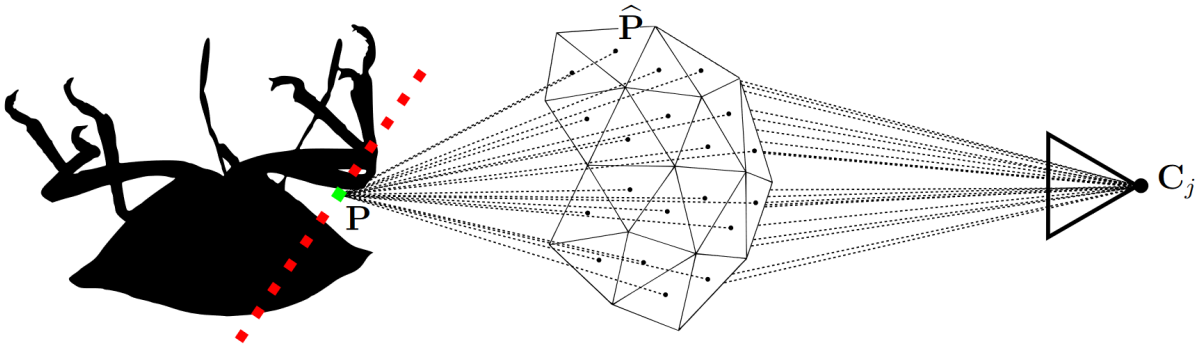


Figure 4 The point of incidence $\overline{\mathbf{P}}_j$ between a 3D point and the center of projection $\mathbf{C}_j$ of the j-th control camera is determined by testing the set of barycenters $\widehat{\mathbf{P}}$ of the triangles of the 3D mesh of the interface that are seen by this camera.

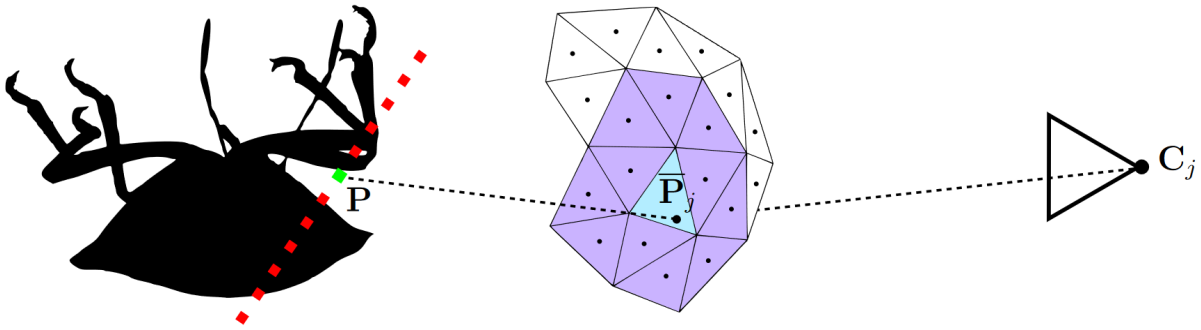


Figure 5 Once the triangle corresponding to the solution of Problem (10) has been identified (triangle indicated in blue), the search for the point of incidence $\overline{\mathbf{P}}_j$ is refined using the method described in Section 3.3. In the case where the solution of this second problem is outside the triangle, a search is performed on the set of adjacent triangles (triangles indicated in purple).

4 Validation on synthetic images

4.1 Cubic interface

We begin by validating our method on a scene featuring a *graphosoma* insect, approximately $30\ mm$ in size, immersed in a refractive cube with an IoR matching that of epoxy resin ($n_2 = 1.56$). The focal length of the camera is $50\ mm$, with an average distance of about $180\ mm$ from the scene. Figure 6 displays two synthetic images (out of a total of 18) of this scene, generated using the ray tracing capabilities of *Blender* software.

Figure 6 Two synthetic images (among 18) of a graphosoma immersed in a cube of epoxy resin. In both cases, the insect is visible through three faces of the cube (only partially, regarding the top face). Due to reflection phenomena, fragments of the object are visible at the boarders of the immersion medium. These image fragments have not been used by our solving method. Source of the 3D model: Digital Archive of Natural History[55].

Figure 7 presents three views of the coloured 3D point cloud reconstructed using our RMVS method. Figure 8 displays this point cloud post-processing, where it has been "cleaned" using the *Connected-component labelling* tool in the *Cloud Compare* software.

Figure 7 3D reconstruction of a graphosoma immersed in a cube of epoxy resin, seen from three angles, obtained by our RMVS solving method from 18 synthetic images such as those in Figure 6.

Figure 9 compares results obtained without considering refraction, utilising three different algorithms: a basic MVS from Equation (5), the Meshroom-proposed MVS pipeline[56] aligned with state-of-the-art algorithms, and neural reconstruction with NeuS2[46]. These methods, not accounting for refraction, fail to interpret the distortions and image duplications of the graphosoma, lead-

Figure 8 Result of Figure 7 after "cleaning" the 3D point cloud by the Connected-component labelling tool of the Cloud Compare software.

ing to poorly reconstructed scenes[1]. Table 1 confirms these shortcomings with high RMSE scores. Conversely, our RMVS solving method, tailored for refraction, demands significantly more computational time: from 5-16 minutes for Figure 9's results to 24 hours for Figure 7's reconstruction (using CPU Intel Xeon Silver 4110 2.10 GHz with all 32 threads for parallel computing), for roughly 500,000 3D points in each instance. Notably, the computation time has been reduced since only those barycenters $\widehat{\mathbf{P}}$ of the mesh triangles (see Figure 4) that project within the insect's silhouette in all the control images are considered.

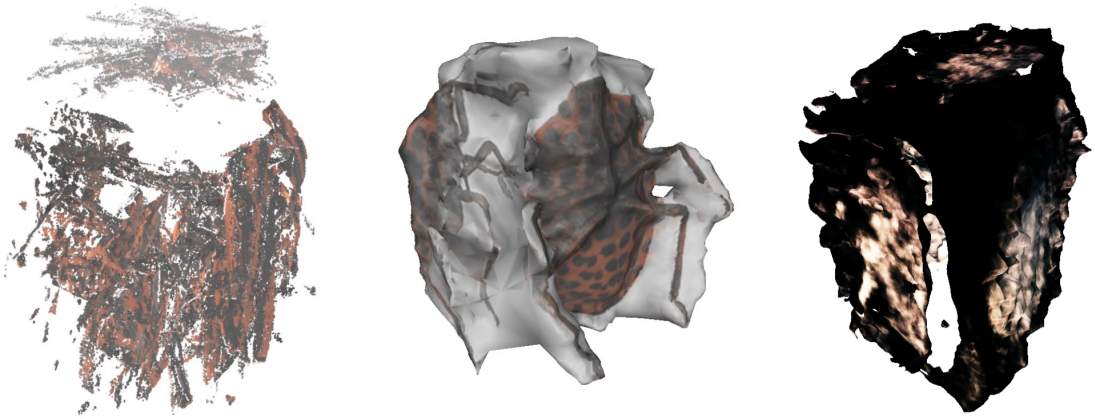


Figure 9 3D reconstruction results without considering refraction. From left to right : a basic MVS derived from Equation (5); the MVS approach by Meshroom[56]; the neural 3D surface reconstruction method NeuS2[46]. The duplication of the graphosoma caused by refraction leads to inaccurate reconstructions.

[1] A comparison with ReNeuS[47], a refraction-inclusive extension of NeuS, would be ideal, but its code is unavailable.

Method	Basic MVS	Meshroom	NeuS2	Ours
RMSE (mm)	6.12	5.44	6.44	**0.57**

Table 1 Root mean square error (RMSE, in mm) comparison between our method and three other methods which do not take refraction into account.

This first example provides insights into our solving method. The 3D reconstruction in Figure 7 is derived from merging eight coloured 3D point clouds, each generated as follows:

- One image is selected as the reference image. Five others serve as control images: in four, the main image of the insect is viewed through the same cube face as in the reference image; in the fifth, it is through an adjacent face.

- For each pixel $\mathbf{p}$ in the reference image, we consider each point $\mathbf{P}$ on the refracted ray from the back-projection of $\mathbf{p}$, for each control image, and for each cube face visible in the control image (up to three per control image). A quartic equation of type (9) is then solved using the Newton-Raphson method.

- After finding a solution, if its projection $\mathbf{p}_j$ in the j-th control image falls inside the insect's silhouette on the relevant face, the similarity between the neighbourhoods of $\mathbf{p}$ and $\mathbf{p}_j$ is computed using a robust estimator, here sum of absolute deviations (SAD). If the SAD is calculable for multiple faces in the j-th control image (up to three), only the smallest value is kept. If no SAD can be calculated, another point $\mathbf{P}$ is tested from a predefined list of 3D points. The chosen point $\mathbf{P}$ is the one that minimises the SAD, it is assigned the colour of the pixel $\mathbf{p}$ in the reference image. If no SAD can be calculated, no 3D point is associated with pixel $\mathbf{p}$.

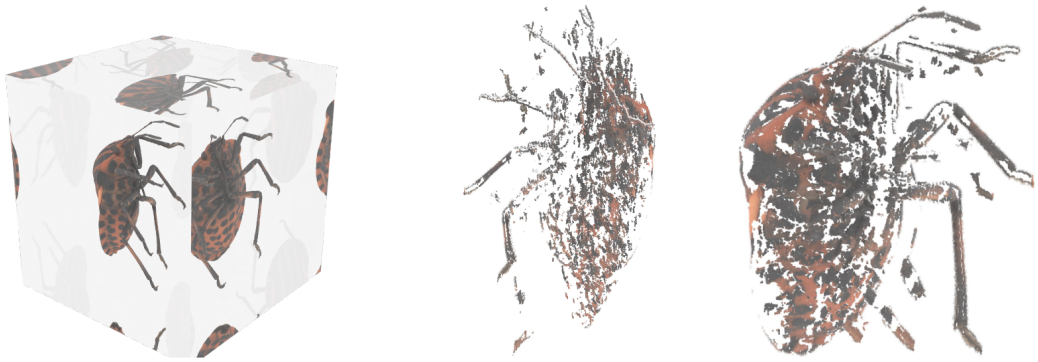
Since all the graphosoma images are synthetic, we can measure the deviations from the ground truth for each of the eight 3D point clouds whose fusion yields the result in Figure 7. Table 2

lists the square root of both the mean and median of these squared deviations. These values are considerably low relative to the scale of the reference 3D model and its distance from the camera, providing quantitative validation for our RMVS solving method.

Face	Front	Front-right	Right	Back-right	Back	Back-left	Left	Front-left	All
RMSE (mm)	0.25	0.48	0.85	0.98	0.40	0.73	0.60	0.65	0.57
RMedSE (mm)	0.15	0.30	0.40	0.40	0.25	0.33	0.33	0.40	0.30

Table 2 Second line: root mean square error (RMSE, in mm) of the eight 3D point clouds whose fusion provides the result of Figure 7. Third line: root median square error (RMedSE, in mm). The last column gives these estimates for all eight 3D point clouds.

Figure 10-a illustrates that the image of a point $\mathbf{P}$ within a refractive medium can produce multiple images, each representing a local minimum of the optical path between $\mathbf{P}$ and the projection center (Fermat principle). Thus, it is feasible to match this image of the insect, effectively applying our RMVS solving method with a single view, as demonstrated in[10,11]. The result, shown in Figures 10-b and 10-c, is rough and incomplete, comprising just a single point cloud. However, this technique differs from other single-view 3D reconstruction methods like shape-from-shading[57], as it relies on the principle of triangulation.



(a) (b) (c)

Figure 10 (a) Example demonstrating the tripling of the graphosoma's image. A 3D point cloud is derived from this single image, selecting the "main" image (right face) as the reference. (b-c) Two perspectives of the 3D point cloud reconstructed by our RMVS solving method, using just this single image.

4.2 Spherical interface

The second experiment involves a graphosoma immersed in an epoxy resin sphere. Figure 11 presents two synthetic images of this setup, alongside images of the graphosoma from identical angles but outside the refractive medium. The notable differences between these image pairs, apart from the magnification effect of the resin sphere acting like a convex lens, are visible in the deformed appearance of the insect's legs and antennae due to refraction. Unlike the images in Figure 6, the images in Figures 11-a and 11-b are not multiplied. They are rendered using ray tracing, approximating the sphere with a triangular mesh of 327,000 faces, generated in Blender from an icosphere with applied subdivisions.

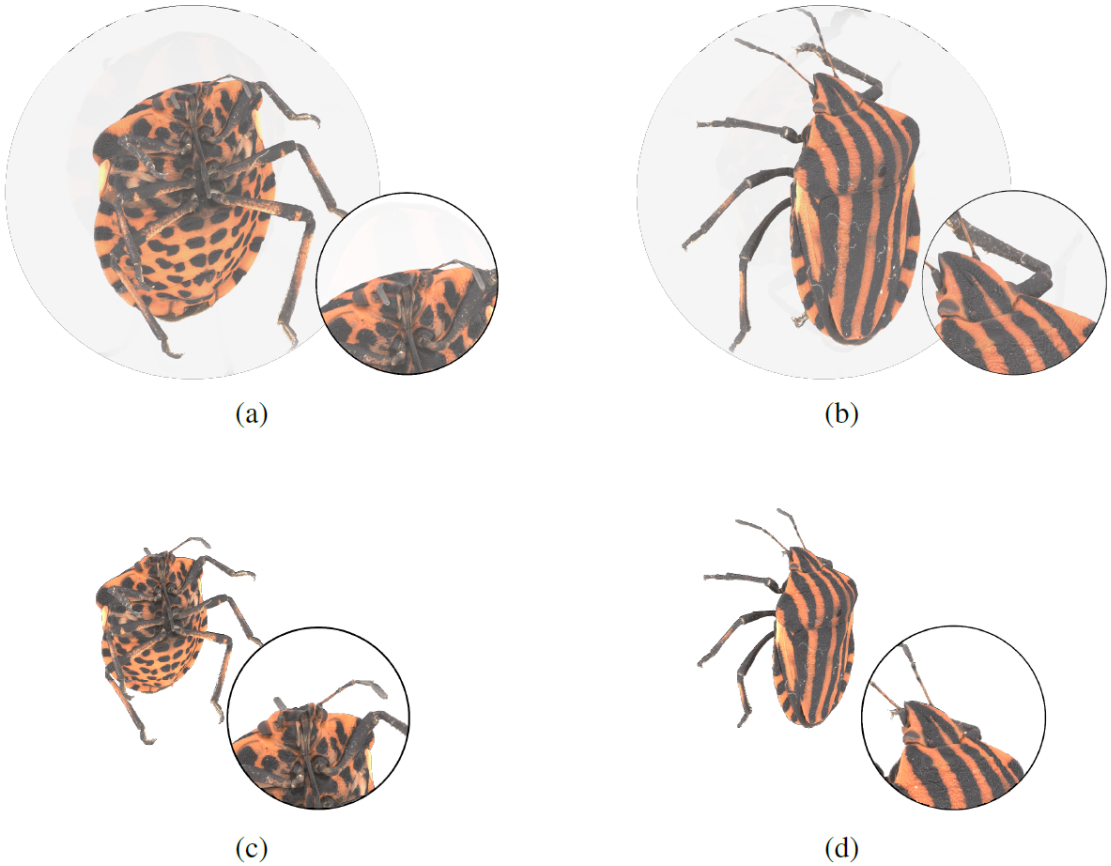


Figure 11 (a-b) Two synthetic images of the graphosoma immersed in an epoxy resin sphere. (c-d) Synthetic images of the graphosoma from the same angles but outside the refractive medium. Along with the magnification effect from the resin's convex shape, the insect's legs and antennae appear deformed.

Figure 12 presents the coloured 3D point cloud from three angles, reconstructed using our RMVS method from 18 images like those in Figures 11-a and 11-b. This cloud is formed by merging eight 3D point clouds. Notably, the legs and antennae of the insect align perfectly across these clouds, and even very fine details are captured. It is important to note that these 3D point clouds are merged with no post-processing, except for cleaning by the Cloud Compare's Connected-component labelling tool. However, the high number of faces on the sphere significantly increases the computation time, from 24 hours for the result of Figure 7 to one week for that of Figure 12.

Figure 12 3D reconstruction of the graphosoma immersed in an epoxy resin sphere, viewed from three angles, obtained with our RMVS solving method with 18 images such as those in Figures 11-a and 11-b. The reconstruction was refined using the Connected-component labelling tool of the Cloud Compare software.

For comparison, Figure 13 shows that when refraction is not considered, MVS struggles to accurately reconstruct the 3D shape, resulting in ghosted legs and antennae. This issue highlights the inconsistency among the eight 3D point clouds.

The choice of interface discretisation scale balances precision with computing time. Table 3 demonstrates the impact of reducing the number of triangular faces in the sphere's 3D mesh (using Cloud Compare's decimation tool), which implies a less precise interface representation. This is assessed through the same two estimators introduced in Section 4.1 (RMSE and RMedSE), along with the percentage of 3D points successfully reconstructed, and the required CPU time.

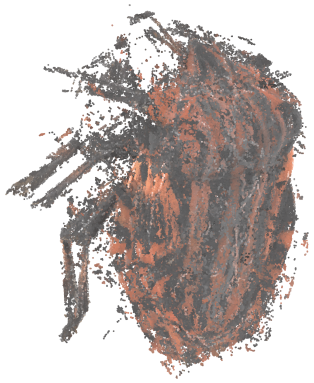


Figure 13 3D reconstruction using MVS from 18 images such as those in Figures 11-a and 11-b, results in ghostly legs and antennae due to inconsistencies among the eight 3D point clouds.

Percentage of faces	100% (327k)	50% (164k)	25% (82k)	10% (33k)	5% (16k)	1% (1.6k)	0.5% (820)
RMSE (mm)	1.77	1.82	1.90	2.03	2.25	3.72	4.32
RMedSE (mm)	0.30	0.33	0.35	0.45	0.57	2.10	3.05
Reconstructed 3D points	98.0%	97.4%	96.2%	93.9%	91.6%	77.8%	69.4%
CPU time (minutes)	498	272	147	49	37	28	23

Table 3 Impact of reducing the number of triangular faces in the sphere's 3D mesh on the 3D reconstruction of the graphosoma using our RMVS solving method (the number of faces used to approximate the sphere is indicated in parentheses).

Figure 14 shows the 3D reconstructions corresponding to Table 3. As anticipated, the first 3D points to "disappear" – those conjugated with pixels p for which no SAD similarity value is calculable – are on the thinnest parts of the 3D model, specifically the legs and antennae.

4.3 Other interfaces

Figure 15 presents two synthetic images of the graphosoma inside a regular dodecahedron made of the same epoxy resin. Our RMVS 3D reconstruction is shown in Figure 16. The deviations from the ground truth are only marginally higher than those in Table 2, despite the images being more challenging for 3D interpretation compared to those in Figure 6. While the RMSE for the entire point cloud increases from $0.57\ mm$ to $1.10\ mm$, the RMedSE rises less significantly, from $0.30\ mm$ to $0.35\ mm$. This higher RMSE value is likely due to substantial image deformation, potentially skewing the SAD estimator's similarity measurement.

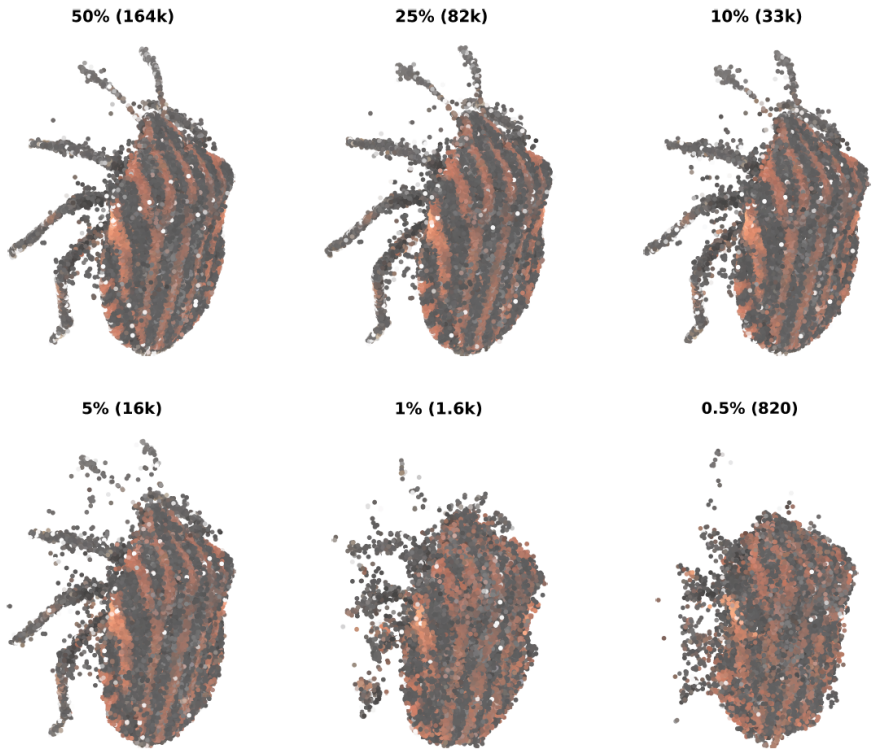


Figure 14 Six 3D reconstructions of the graphosoma immersed in an epoxy resin sphere, illustrating the effect of reducing the percentage of triangular faces of the sphere used in our RMVS solving method (refer to Table 3). Figure 12 displays the outcome when all 327,000 faces are utilised.

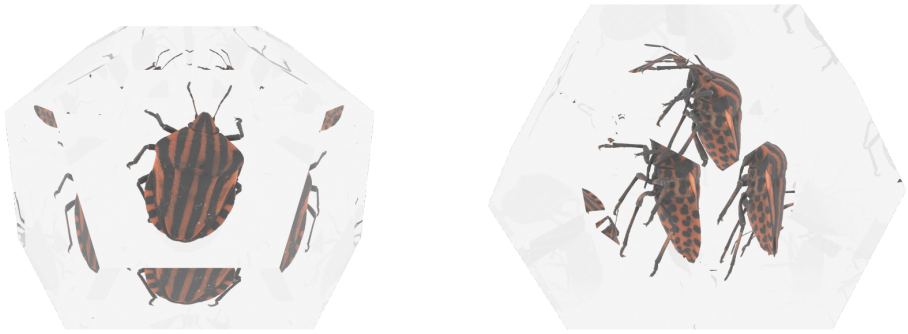


Figure 15 Two images of the graphosoma immersed in an epoxy resin regular dodecahedron.

Figure 17 shows that images can undergo even more distortion with a block of any convex shape. The 3D reconstruction achieved by our RMVS solving method, as shown in Figure 18, remains true to the original form but is slightly less precise than the reconstruction in Figure 7. This reduced precision is due to some grazing rays, where the angle i_2 in the Snell-Descartes law (7) approaches $\pi/2$. Consequently, since the derivative of the $\arcsin$ function tends towards infinity

Figure 16 3D reconstruction of the graphosoma immersed in an epoxy resin regular dodecahedron, viewed from three angles. This was created using our RMVS solving method from 18 images such as those in Figure 15, followed by refinement using the Connected-component labelling tool of the Cloud Compare software.

at 1, this results in calculation inaccuracies for the angle i_1 in Equation (7). In contrast, Figure 19 illustrates that neglecting refraction in the reconstruction process yields a result resembling a random 3D point cloud.

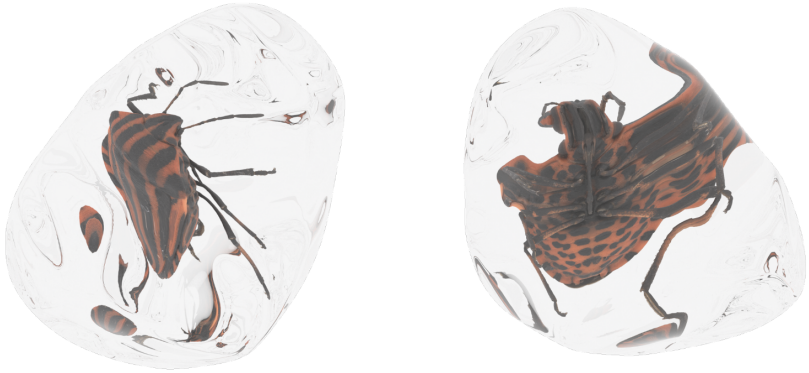


Figure 17 Two images of the graphosoma immersed in a convex block of epoxy resin.

Figure 18 3D reconstruction of the graphosoma immersed in a convex block of epoxy resin, seen from three angles, using our RMVS solving method from 18 images such as those in Figure 17.

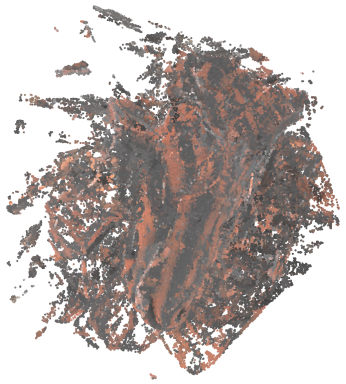


Figure 19 3D reconstruction by MVS, from 18 images such as those in Figure 17: the result resembles a random 3D point cloud.

In this section, we validated our RMVS solving method only on synthetic images, but purposely. Indeed, to be able to process real images, several additional data are necessary, in addition to the images themselves and the intrinsic parameters of the camera: the shape of the interface (with more or less precision, see Table 3), the poses of the camera and the IoR of the transparent medium.

5 Implementation on real images

The primary challenge in applying our 3D reconstruction method to real images lies in estimating camera poses. While the refractive structure-from-motion method suggested in[53] is an option, it requires prior knowledge of the medium's IoR, which is one of the unknown factors. Additionally, since recovering the 3D shape of the interface is essential, we propose in Section 5.1 a simultaneous estimation method for both camera poses and the interface 3D shape. This approach relies on multi-view matching of polyhedron vertices detected in the images and does not rely on the IoR. Consequently, the IoR can be determined a posteriori, as we will discuss in Section 5.2.

5.1 Estimating the camera poses and the interface 3D shape

In this subsection, we detail a method for acquiring the camera poses and the 3D shape of the interface in a shared 3D frame. This involves fixing the camera opposite the object positioned on a rotating table, simulating camera movement around the object. Figures 20-a and 21-a illustrate this setup.

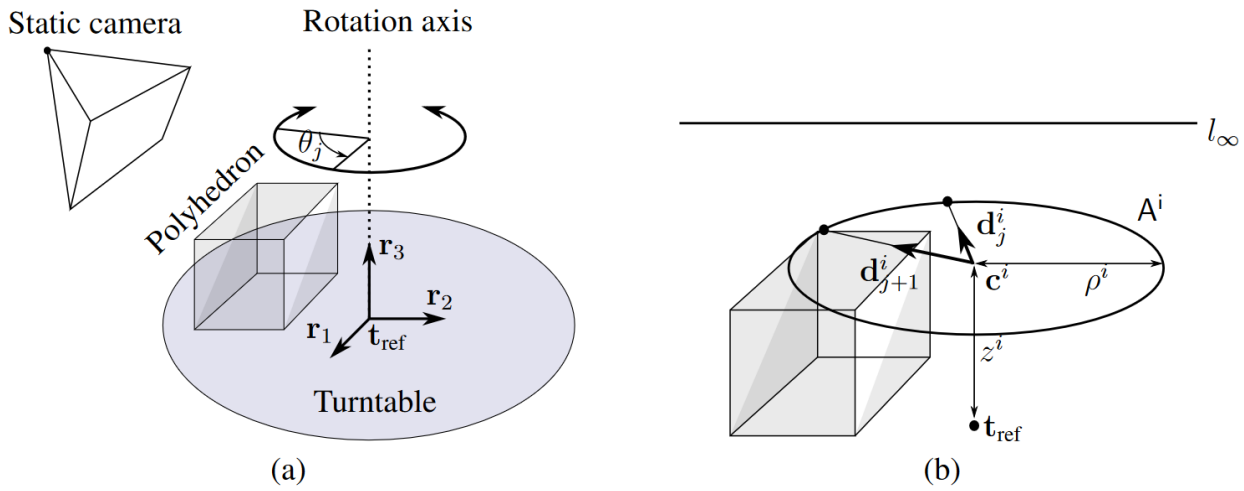


Figure 20 (a) The acquisition setup involves placing a polyhedron on a turntable and capturing views with a static camera. The origin $\mathbf{t}_{\text{ref}}$ is at the intersection of the table and its rotation axis. The rotation matrix $\mathbf{R}_{\text{ref}}$, having columns $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$, defines the turntable's pose relative to the camera frame. (b) In consecutive images j and $j+1$, a vertex at a distance ρ^i from the rotation axis, oriented along $\mathbf{d}^i_j$ and $\mathbf{d}^i_{j+1}$, belongs to an ellipse of equation $\mathbf{x}^\top \mathsf{A}^i \mathbf{x} = 0$ in the image plane. The imaged center $\mathbf{c}^i$ of this ellipse satisfies the pole-polar relation $\mathbf{c}^i = [\mathsf{A}^i]^{-1}\mathbf{l}_\infty$.

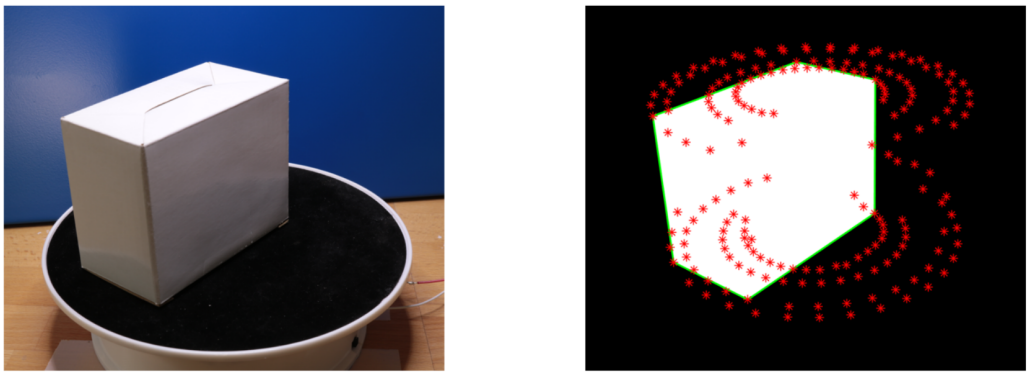


Figure 21 (a) One of 40 images depicting a parcel on a turntable. In all views, the silhouettes of the parcel, treated as a convex polyhedron, are extracted. (b) The collection V of all silhouette vertices is shown in red.

We could have concurrently estimated the camera poses and the 3D shape of the interface using shape-from-silhouettes, a technique independent of the IoR of the medium. However, this method struggles with arbitrary shapes, as it computes an enclosing volume by intersecting silhouette back-projections. For satisfactory accuracy, an infinite number of poses are ideal, unless we limit to polyhedral interfaces with a few vertices. For insects in amber, as mentioned in Section 1, this is feasible by shaping the amber into a polyhedron with multiple planar faces.

We thus consider a scene with a convex polyhedron of q vertices on a turntable. The vertices' coordinates $\mathbf{X}^i \in \mathbb{R}^3$, $i \in \{1, \ldots, q\}$, are in a 3D frame $\mathcal{R}_{\text{ref}}$ affixed to the turntable. The origin of $\mathcal{R}_{\text{ref}}$ lies at the table's supporting plane and rotation axis intersection, with its first two axes defining the plane and the third as an upward normal vector. A static camera with known intrinsics captures r views of this scene. The homogeneous coordinate vector $\mathbf{x}_j^i \in \mathbb{R}^3$ of the j-th image, $j \in \{1, \ldots, r\}$, of vertex $\mathbf{X}^i$ satisfies the equation:

$$\mathbf{x}_j^i \sim \mathsf{P}_j \begin{bmatrix} \mathbf{X}^i \\ \\ 1 \end{bmatrix} \tag{11}$$

with the perspective projection matrix P_j corresponding to the j-th view defined as:

$$\mathsf{P}_j = \mathbf{K} \, [\, \mathbf{R}_{\text{ref}} \, \mathbf{R}_j \mid \mathbf{t}_{\text{ref}} \,] \tag{12}$$

In (12), $\mathbf{K}$ represents the calibration matrix. The transformation from $\mathcal{R}_{\text{ref}}$ to the camera frame is

specified by $(\mathbf{R}_{\text{ref}}, \mathbf{t}_{\text{ref}})$. The matrix $\mathbf{R}_j$ indicates the table's rotation by an angle θ_j around its axis:

$$\mathbf{R}_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 \\ \sin\theta_j & \cos\theta_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

In homogeneous Cartesian coordinates $\mathbf{x} = [x, y, 1]^\top$, an ellipse is represented as $\mathbf{x}^\top \mathsf{A}\, \mathbf{x} = 0$, where A is a symmetric 3×3 matrix under suitable conditions on its coefficients.

The first step is to create a polygonal silhouette for each view and is obtained by simple operations: background subtraction from a reference image, thresholding, morphological processing, extraction, and simplification of the convex hull to get the silhouette vertices. We then assemble the collection V of all these vertices. An example of extracted silhouette vertices V, highlighted in red, is shown in Figure 21-b.

The second step requires robustly partitioning V into subsets on common ellipses, representing *parallel circular trajectories* in 3D space. The partition size, corresponding to the polyhedron's vertices, is unknown. A partitioning solution is detailed in[2], utilising the parallelism of vertices' trajectories. The *images of circular points* (ICP)[58] of the turntable, two complex conjugate vectors in $\mathbb{C}^3$ denoted as $\mathbf{h}_1 \pm \mathrm{i}\mathbf{h}_2$, are estimated along with correspondences. Details can be found in[2].

With the ICP $\mathbf{h}_1 \pm \mathrm{i}\mathbf{h}_2$, correspondences $\{\mathbf{x}_j^i\}$, and calibration matrix K known, the problem is to determine the vertices' positions and the polyhedron's poses in the camera frame. Specifically, this includes calculating the rotation matrix $\mathbf{R}_{\text{ref}}$, the translation vector $\mathbf{t}_{\text{ref}}$, the 3D coordinates of the vertices $\{\mathbf{X}^i\}_{i\in\{1,\ldots,q\}}$, and the angles $\{\theta_j\}_{j\in\{1,\ldots,r\}}$. Matrix $\mathbf{R}_{\text{ref}}$ and vector $\mathbf{t}_{\text{ref}}$ are computed using the method from[59].

The rotation angle θ_j of the turntable in view number j is measured from a reference position θ_1 as follows:

$$\theta_j = \sum_{k=1}^{j-1} \theta_{k,k+1} \tag{14}$$

where $\theta_{k,k+1}$ represents the rotation angle between two consecutive acquisitions k and $k+1$. Its value is determined as the median cosine of the estimated angles from the visible vertices:

$$\theta_{k,k+1} = \operatorname{acos}\left(\underset{i \in \mathcal{D}_k}{\operatorname{median}} \left\{ {\mathbf{d}_k^i}^\top \mathbf{d}_{k+1}^i \right\}\right) \tag{15}$$

where $\mathcal{D}_k \subset \{1, \ldots, q\}$ represents the set of vertex indices detected in both the k^{th} and $(k+1)^{\text{th}}$ images. The unit vectors $\mathbf{d}_k^i$ and $\mathbf{d}_{k+1}^i$ point towards the images by H^{-1} of the corresponding points $\mathbf{x}_k^i$ and $\mathbf{x}_{k+1}^i$, where $\mathsf{H} = [\,\mathbf{h}_1\ \mathbf{h}_2\ *]$, in the sequential views k and $k+1$, specifically:

$$\mathbf{d}_k^i = \frac{f(\mathsf{H}^{-1}\mathbf{x}_k^i) - f(\mathsf{H}^{-1}\mathbf{c}^i)}{\|f(\mathsf{H}^{-1}\mathbf{x}_k^i) - f(\mathsf{H}^{-1}\mathbf{c}^i)\|} \tag{16}$$

and likewise for $\mathbf{d}_{k+1}^i$. In (16), $f([u,v,w]^\top) = [u/w, v/w]^\top$, and $\mathbf{c}^i$ is the homogeneous coordinate vector of the image of the trajectory's center, assumed circular, of the vertex number i, and derived from the pole-polar relation $\mathbf{c}^i = [\mathsf{A}^i]^{-1}\mathbf{l}_\infty$. Here $\mathbf{l}_\infty$ is the vanishing line vector of the table plane, and is the cross-product $\mathbf{l}_\infty = \mathbf{h}_1 \times \mathbf{h}_2$, and A^i the matrix of the ellipse image of vertex number i's trajectory (see Figure 20-b). The table rotation during acquisition is assumed to be counterclockwise. The $\theta_{k,k+1}$ values are supposed to be between 0 and 180 degrees.

At this point, all camera poses are known, and the 3D coordinates of the vertices $\{\mathbf{X}^i\}$ are obtained by triangulating the correspondences $\{\mathbf{x}_j^i\}$. Both are further refined through a bundle adjustment minimising the Euclidean distances between the correspondences and their reprojections.

5.2 Validation on real images

Estimating the index of refraction can typically be done using a dedicated instrument known as a *refractometer*. However, we suggest an alternative estimation method in this subsection, leveraging the joint estimation of camera poses and the 3D shape of the interface, as outlined in the previous subsection. Specifically, our RMVS solving method, detailed in Section 3 and tested on synthetic images in Section 4, can be applied with varying IoR values, ideally within a range close to its "plausible" value. The challenge lies in identifying a sufficiently discriminating criterion for this estimation. As illustrated in Figures 22 and 23, the number of effectively reconstructed 3D points serves as such a criterion, since it shows the maximum number of points for the exact IoR and the lowest Chamfer distance score for the associated reconstruction.

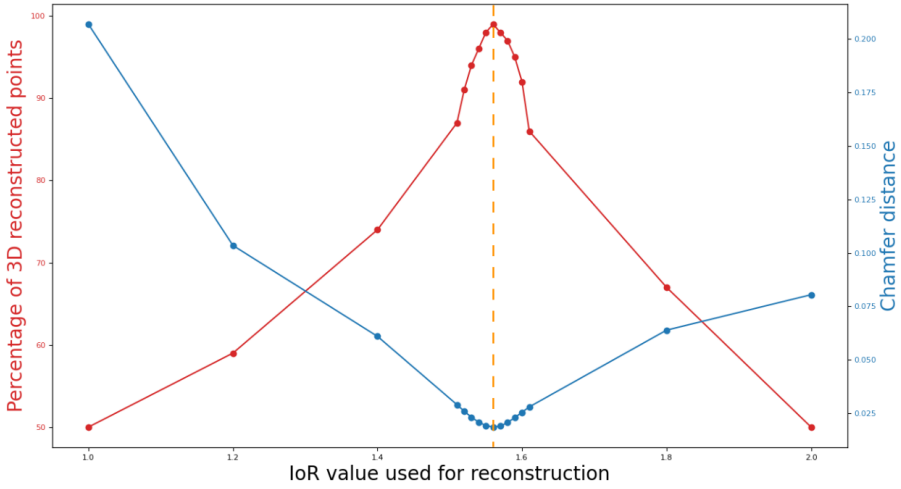


Figure 22 Percentage of reconstructed 3D points (red) and Chamfer distance (blue) variation with the index of refraction (IoR) of the refractive medium surrounding the graphosoma. When simulating the images, the IoR used ($n_2 = 1.56$) aligns exactly with the peak of the red curve and the lowest CD, validating our proposed criterion.

We can now apply the complete RMVS solving pipeline to real data, provided the interface is polyhedral. Figures 24 and 25 display tests conducted on two epoxy-resin parallelepipeds, containing a beetle and a grasshopper, respectively.

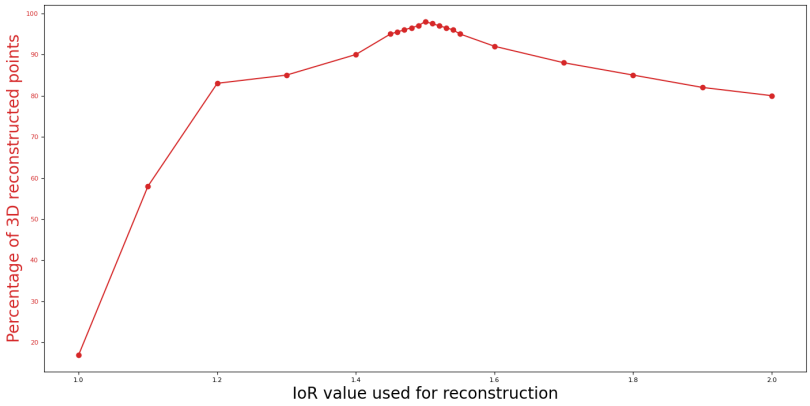


Figure 23 Evolution of the percentage of reconstructed 3D points, in function on the IoR of the refractive medium in which the real beetle from Figure 25 is immersed. The maximum of this curve gives us an estimate of the IoR equal to $n_2 = 1.50$.

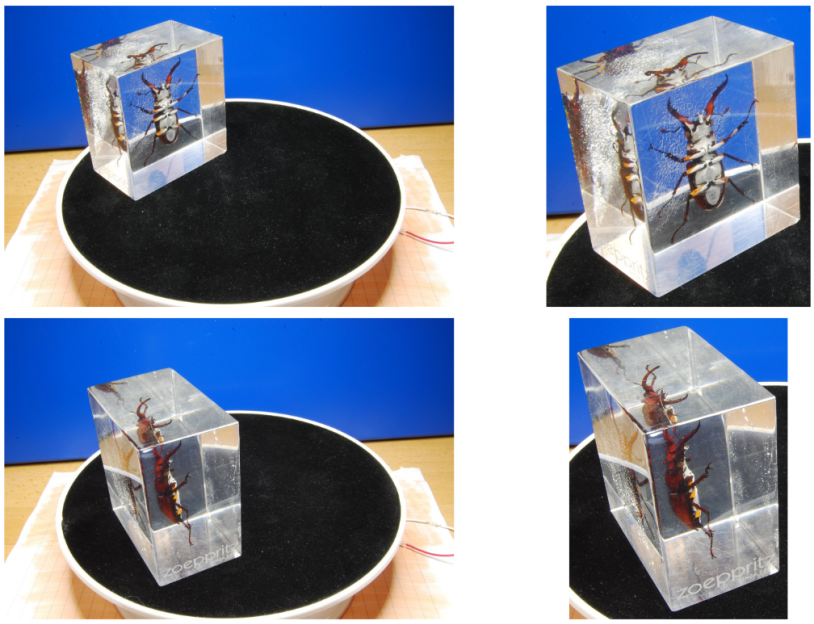


Figure 24 Two real images of a beetle immersed in a parallelepipedic block of resin, placed on a turntable, and zooms on the block.

The 3D reconstructions of these two insects, shown in Figures 26 and 27, reveal a noticeably better reconstruction of the beetle compared to the grasshopper. This disparity primarily stems from the resin block containing the grasshopper, which fails to fully meet the assumptions underlying our RMVS solving method. Firstly, one of the block's faces is not as planar as required. Secondly, the resin exhibits layered structure visible to the naked eye, indicating that light rays

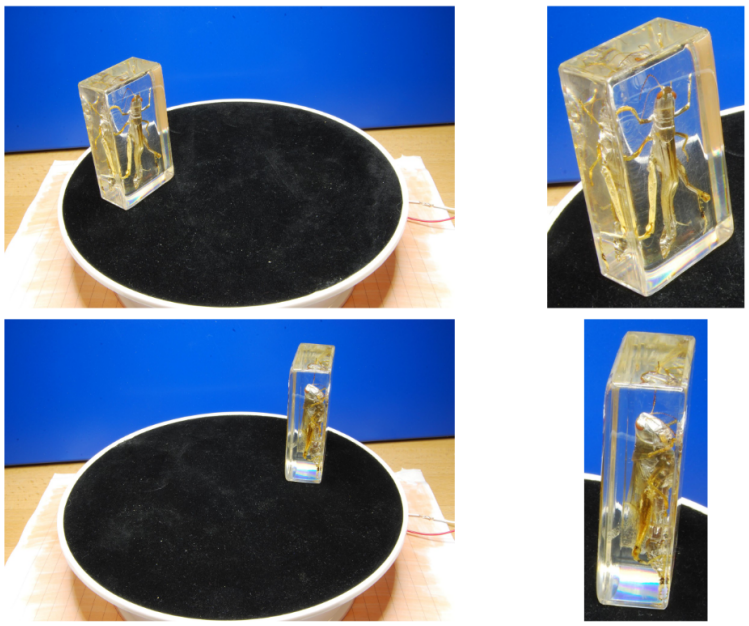


Figure 25 Two real images of a grasshopper immersed in a parallelepipedic block of resin, placed on a turntable, and zooms on the block.

within the refractive medium may not travel in perfectly straight lines. Additionally, a significant distinction between the results in Figures 26 and 27 lies in the insects themselves. Certain parts of the grasshopper's body appear somewhat translucent, challenging a fundamental premise of the MVS technique and its variants, which is the assumption that the surface being reconstructed should be opaque and Lambertian.

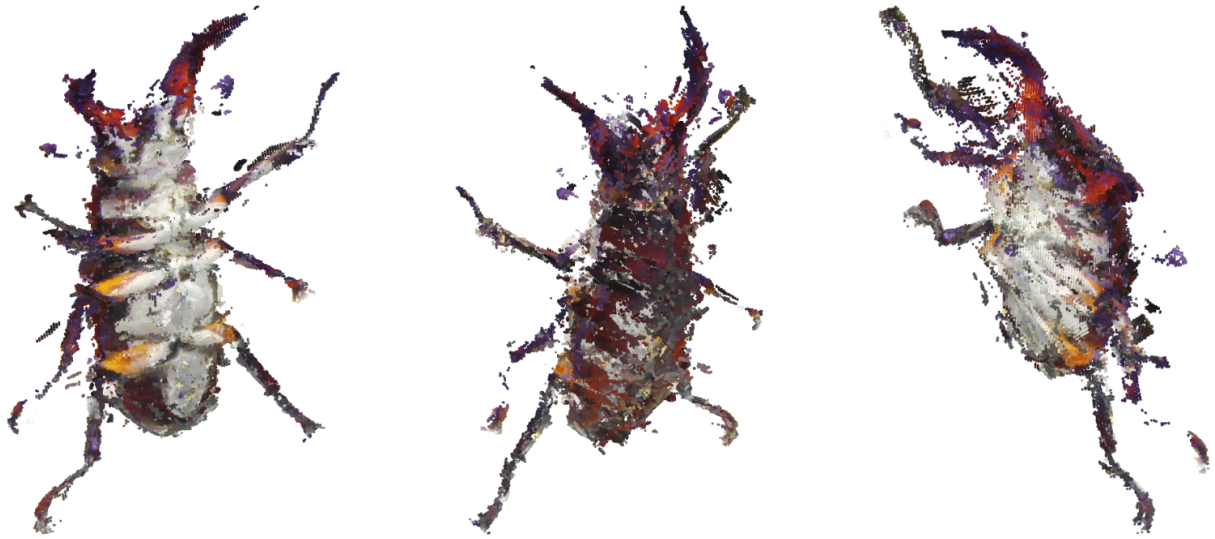


Figure 26 3D reconstruction of the beetle from 24 images such as those in Figure 24, using our RMVS method.

Figure 27 3D reconstruction of the grasshopper from 24 images such as those in Figure 25, using our RMVS method.

A final experiment with the beetle images aimed to qualitatively assess how using an incorrect IoR value affects the reconstruction outcome. Figure 28 displays two 3D reconstructions of the beetle, each derived using different IoR values: the right image, produced with a slightly overvalued IoR ($n_2' = 1.56$), is noticeably less accurate than the left image, where the IoR ($n_2 = 1.50$) was determined using the previously described method (refer to Figure 23).

Figure 28 Comparison of our RMVS method, tested on the same 24 real images of the beetle (see Figure 24). We used either the IoR value estimated by the method illustrated in Figure 23 ($n_2 = 1.50$), or a slightly overvalued IoR ($n_2' = 1.56$). The first result is obviously more accurate.

6 Conclusion and perspectives

In this paper, we adapted the MVS technique for objects immersed in a refractive medium. Given that refraction distorts images, it is crucial to model light ray paths accordingly. We introduced a fully discrete RMVS solving method, with promising initial results on real data, despite several challenges before it becomes a practical tool for entomologists.

A future direction involves assessing the RMVS method's robustness against imperfect knowledge of interface geometry, such as non-planar polyhedron faces. The use of UV, IR and polarized lights could also help us to constrain the interface geometry and to reduce some refraction/reflection effects. Another area for development is automating the detection of silhouettes within the refractive medium. Neural methods, as suggested by[60], could be a solution.

Furthermore, methods using differentiable rendering, like ReNeuS, are increasingly important. We were unfortunately unable to test ReNeuS as its code is not publicly available (and it does consider only boxed-shaped media). However, such approaches remain a short-term goal, whether to solve the RMVS problem addressed in this paper or to solve photometric stereo under refraction[36].

A longer-term goal is to develop a pipeline for acquiring and processing data, particularly prehistoric insects trapped in amber. Overcoming numerous challenges is necessary, as the poor result in Figure 27 is due to both the resin block and the contained object not fully meeting our RMVS method's assumptions. The pipeline needs to be robust against predictable flaws, particularly when the index of refraction is not uniform. Additionally, even under the Lambertian assumption, colouration in the refractive medium can alter a 3D point's appearance across images due to varying light travel distances. Focus blur, a small-scale challenge we have overlooked also needs consideration. Addressing these factors should enhance the quality of our results.

Acknowledgements. Robin Bruneau's doctoral student fellowship is funded by the Danish project UCPH Data+ PHYLORAMA. Baptiste Brument's doctoral student fellowship is funded by the French Ministry of Higher Education and Research. This work was partly funded by the French National Research Agency through the LabCom project ALICIA-Vision and the Inclusive Museum Guide project (ANR-20-CE38-0007).

Code, Data, and Materials Availability. Our real/synthetic data and code will be available on demand.

References

1 M. Cassidy, J. Mélou, Y. Quéau, *et al.*, "Refractive Multi-view Stereo," in *Proceedings of the International Conference on 3D Vision*, (2020).

2 B. Brument, L. Calvet, R. Bruneau, *et al.*, "A Shape-from-silhouette Method for 3D-reconstruction of a Convex Polyhedron," in *Proceedings of the Quality Control by Artificial Vision Conference*, (2023).

3 H. G. Maas, "New developments in multimedia photogrammetry," in *Optical 3-D Measurement Techniques III*, (1995).

4 T. Łuczyński, M. Pfingsthorn, and A. Birk, "Image rectification with the pinax camera model in underwater stereo systems with verged cameras," in *OCEANS 2017*, 1–7 (2017).

5 P. Agrafiotis, K. Karantzalos, A. Georgopoulos, *et al.*, "Correcting image refraction: Towards accurate aerial image-based bathymetry mapping in shallow waters," *Remote Sensing* **12**(2), 322 (2020).

6 X. Wu and X. Tang, "Accurate binocular stereo underwater measurement method," *International Journal of Advanced Robotic Systems* **16**(5) (2019).

7 P. Agrafiotis, D. Skarlatos, A. Georgopoulos, *et al.*, "DepthLearn: learning to correct the refraction on point clouds derived from aerial imagery for accurate dense shallow water bathymetry based on SVMs-fusion with LiDAR point clouds," *Remote Sensing* **11**(19), 2225 (2019).

8 K. Ichimaru and H. Kawasaki, "Underwater Stereo Using Refraction-Free Image Synthesized From Light Field Camera," in *Proceedings of the IEEE International Conference on Image Processing*, 1039–1043 (2019).

9 C. Zhang, X. Zhang, D. Tu, *et al.*, "On-site calibration of underwater stereo vision based on light field," *Optics and Lasers in Engineering* **121**, 252–260 (2019).

10 D. H. Lee, I.-S. Kweon, and R. Cipolla, "A biprism-stereo camera system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **1** (1999).

11 A. Yamashita, Y. Shirane, and T. Kaneko, "Monocular Underwater Stereo – 3D Measurement Using Difference of Appearance Depending on Optical Paths," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3652–3657 (2010).

12 Z. Chen, K.-Y. K. Wong, Y. Matsushita, *et al.*, "Depth from refraction using a transparent medium with unknown pose and refractive index," *International Journal of Computer Vision* **102**(1–3), 3–17 (2013).

13 C. Gao and N. Ahuja, "A Refractive Camera for Acquiring Stereo and Super-resolution Images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2316–2323 (2006).

14 N. J. W. Morris, “Image-based water surface reconstruction with refractive stereo,” Master’s thesis, Department of Computer Science, University of Toronto (2004).

15 N. J. W. Morris and K. N. Kutulakos, “Dynamic Refraction Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(8), 1518–1531 (2011).

16 M. Ben-Ezra and S. K. Nayar, “What does motion reveal about transparency?,” in *Proceedings of the IEEE International Conference on Computer Vision*, **2**, 1025–1032 (2003).

17 Z. Li, Y.-Y. Yeh, and M. Chandraker, “Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1262–1271 (2020).

18 M. Shao, C. Xia, D. Duan, *et al.*, “Polarimetric inverse rendering for transparent shapes reconstruction,” *IEEE Transactions on Multimedia* **26**, 7801–7811 (2024).

19 T. Murase, M. Tanaka, T. Tani, *et al.*, “A Photogrammetric Correction Procedure for Light Refraction Effects at a Two-Medium Boundary,” *Photogrammetric Engineering and Remote Sensing* **9**(8), 1129–1136 (2008).

20 A. S. Woodget, J. T. Dietrich, and R. T. Wilson, “Quantifying below-water fluvial geomorphic change: The implications of refraction correction, water surface elevations, and spatially variable error,” *Remote Sensing* **11**(20), 2415 (2019).

21 B. Cao, R. Deng, and S. Zhu, “Universal algorithm for water depth refraction correction in through-water stereo remote sensing,” *International Journal of Applied Earth Observation and Geoinformation* **91**, 102108 (2020).

22 P. Sturm, “Multi-view geometry for general camera models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **1**, 206–212 (2005).

23 V. Chari and P. Sturm, "Multi-View Geometry of the Refractive Plane," in *Proceedings of the British Machine Vision Conference*, 1–11 (2009).

24 C. Chen, H. Wang, Z. Zhang, *et al.*, "Three-dimensional reconstruction from a fringe projection system through a planar transparent medium," *Optics Express* **30**(19), 34824–34834 (2022).

25 A. Jordt-Sedlazeck and R. Koch, "Refractive calibration of underwater cameras," in *Proceedings of the European Conference on Computer Vision*, 846–859 (2012).

26 A. Jordt-Sedlazeck, D. Jung, and R. Koch, "Refractive Plane Sweep for Underwater Images," in *Proceedings of the German Conference on Pattern Recognition*, 333–342 (2013).

27 A. Jordt, K. Köser, and R. Koch, "Refractive 3D reconstruction on underwater images," *Methods in Oceanography* **15–16**, 90–113 (2016).

28 S. Haner and K. Åström, "Absolute pose for cameras under flat refractive interfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1428–1436 (2015).

29 M. Castillón, A. Palomer, J. Forest, *et al.*, "State of the Art of Underwater Active Optical 3D Scanners," *Sensors* **19**(23), 5161 (2019).

30 M. Alterman and Y. Y. Schechner, "3D in natural random refractive distortions," in *Three-Dimensional Imaging, Visualization, and Display 2016*, **9867**, 64–75, SPIE (2016).

31 M. Alterman, Y. Y. Schechner, and Y. Swirski, "Triangulation in Random Refractive Distortions," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(3), 603–616 (2017).

32 J. Xiong and W. Heidrich, "In-the-wild single camera 3D reconstruction through moving water surfaces," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12558–12567 (2021).

33 C. Tsiotsios, M. E. Angelopoulou, T.-K. Kim, *et al.*, "Backscatter compensated photometric stereo with 3 sources," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2251–2258 (2014).

34 S. G. Narasimhan, S. K. Nayar, B. Sun, *et al.*, "Structured light in scattering media," in *Proceedings of the IEEE International Conference on Computer Vision*, **1**, 420–427 (2005).

35 H. Fan, L. Qi, Y. Ju, *et al.*, "Refractive laser triangulation and photometric stereo in underwater environment," *Optical Engineering* **56**(11), 113101–113101 (2017).

36 Y. Quéau, R. Bruneau, J. Mélou, *et al.*, "On Photometric Stereo in the Presence of a Refractive Interface," in *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision*, 691–703 (2023).

37 S. Liu, T. Li, W. Chen, *et al.*, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7708–7717 (2019).

38 J. Lyu, B. Wu, D. Lischinski, *et al.*, "Differentiable refraction-tracing for mesh reconstruction of transparent objects," *ACM Transactions on Graphics* **39**(6), 1–13 (2020).

39 J. Munkberg, J. Hasselgren, T. Shen, *et al.*, "Extracting triangular 3d models, materials, and lighting from images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8280–8290 (2022).

40 M. Nimier-David, D. Vicini, T. Zeltner, *et al.*, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Transactions on Graphics* **38**(6), 1–17 (2019).

41 W. Jakob, S. Speierer, N. Roussel, *et al.*, "Dr. jit: A just-in-time compiler for differentiable rendering," *ACM Transactions on Graphics* **41**(4), 1–19 (2022).

42 K. Yan, C. Lassner, B. Budge, *et al.*, "Efficient estimation of boundary integrals for path-space differentiable rendering," *ACM Transactions on Graphics* **41**(4), 1–13 (2022).

43 M. Bemana, K. Myszkowski, J. Revall Frisvad, *et al.*, "Eikonal fields for refractive novel-view synthesis," in *Proceedings of the ACM SIGGRAPH Conference*, 1–9 (2022).

44 T. Fujitomi, K. Sakurada, R. Hamaguchi, *et al.*, "LB-NERF: Light Bending Neural Radiance Fields for Transparent Medium," in *Proceedings of the IEEE International Conference on Image Processing*, 2142–2146 (2022).

45 P. Wang, L. Liu, Y. Liu, *et al.*, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689* (2021).

46 Y. Wang, Q. Han, M. Habermann, *et al.*, "NeuS2: Fast Learning of Neural Implicit Surfaces for Multi-view Reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2023).

47 J. Tong, S. Muthu, F. A. Maken, *et al.*, "Seeing Through the Glass: Neural 3D Reconstruction of Object Inside a Transparent Container," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12555–12564 (2023).

48 Y. Furukawa and J. Ponce, "Accurate, Dense, and Robust Multiview Stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(8), 1362–1376 (2010).

49 L. Kang, L. Wu, and Y.-H. Yang, "Two-View Underwater Structure and Motion for Cameras under Flat Refractive Interfaces," in *Proceedings of the European Conference on Computer Vision*, 303–316 (2012).

50 A. Agrawal, S. Ramalingam, Y. Taguchi, *et al.*, "A theory of multi-layer flat refractive geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3346–3353 (2012).

51 Y. Furukawa and C. Hernández, "Multi-View Stereo: A Tutorial," *Foundations and Trends in Computer Graphics and Vision* **9**(1-2), 1–148 (2015).

52 M. Goesele, B. Curless, and S. M. Seitz, "Multi-View Stereo Revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2402–2409 (2006).

53 F. Chadebecq, F. Vasconcelos, R. Lacher, *et al.*, "Refractive Two-View Reconstruction for Underwater 3D Vision," *International Journal of Computer Vision* **128**(5), 1101–1117 (2020).

54 E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik* **1**, 269–271 (1959).

55 Y. Nehmé, J. Delanoy, F. Dupont, *et al.*, "Textured mesh quality assessment: Large-scale dataset and deep learning-based quality metric," *ACM Transactions on Graphics* **42**(3), 1–20 (2023).

56 C. Griwodz, S. Gasparini, L. Calvet, *et al.*, "AliceVision Meshroom: An open-source 3D reconstruction pipeline," in *Proceedings of the 12th ACM Multimedia Systems Conference*, 241–247 (2021).

57 J.-D. Durou, M. Falcone, and M. Sagona, "Numerical Methods for Shape-from-shading: A

New Survey with Benchmarks," *Computer Vision and Image Understanding* **109**(1), 22–43 (2008).

58 R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Second ed. (2004).

59 P. Sturm, "Algorithms for Plane-Based Pose Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2000).

60 A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026 (2023).