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, 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[a], Matthew Cassidy[a], Jean Mélou[a], Yvain Quéau[c], Jean-Denis Durou[a], François Lauze[b]

[a] *IRIT, UMR CNRS 5505, Toulouse, France*
[b] *DIKU, Copenhagen, Denmark*
[c] *Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France*

ABSTRACT

In this article we show how to extend the multi-view stereo 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.

1. Introduction

Many Natural History Museums have collections of fragile specimens immersed in a transparent medium, for instance a prehistoric insect trapped in a block of amber, or a small animal conserved in a solution of formaldehyde or alcohol in a glass jar (see Figure 1), which constitute an invaluable source of information for evolutionary scientists.

Exploitation of these resources necessitates "see through" 3D estimation techniques. Computerised tomography (CT) provides a 3D reconstruction, however it requires special CT scanners that can handle objects of various sizes. This is expensive and time consuming, and may be feasible for a limited amount of samples but out of reach for large collections (thousands of amber pieces, or linear kilometers of specimens in jars). Photogrammetric 3D scanning is a reasonable alternative, which however requires overcoming several difficulties.

*Corresponding author
e-mail: rb@di.ku.dk (Robin Bruneau)

Figure 1. Left: prehistoric beetle trapped in amber (seen under a microscope). Right: reptiles specimens in jars (image from the Natural History Museum in Copenhagen).

First of all, as described by Snell's law, if the *refractive* medium deflects the trajectory of light rays that pass through it, the shape of the *interface*, i.e. the separation surface between air and this medium, has a strong influence on the appearance of the object to be reconstructed. In the example on the left in Figure 1, it is not forbidden to cut the amber block to give it a particular shape, as long as this does not alter the insect. The shape of the refractive interface will thus be assumed to be known. However, unlike most existing techniques, we do not restrict our attention to the planar case.

Refraction can cause other difficulties. Because it is wavelength dependent, it causes *light dispersion*. In addition, an inhomogeneous transparent medium may have a spatially-varying refractive index, resulting in light rays that are not straight. Taking a varying refractive index into account does not present an intractable difficulty, but estimating this index at each point in the medium is in itself a difficult problem. We will therefore assume that the refractive medium is *homogeneous* and that the refractive index varies little with wavelength.

Contributions. We introduce a general 3D reconstruction method for objects trapped inside a homogeneous refractive medium with arbitrary, known interface, from a set of calibrated multi-view stereo images. The paper is organized as follows. In Section 2, we review the main studies taking refraction into account. We show in Section 3 how to extend the multi-view stereo technique in the presence of an interface. The main difficulty of this extension consists in predicting the projection in an image of a 3D point located in the refractive medium. A first set of tests on synthetic images is conducted in Section 4 to validate this method. In Section 5, we show how to validate it on real data, which requires additional information about the 3D scene. Finally, we conclude in Section 6 by mentioning some possible extensions. The end-to-end pipeline described in this paper extends two of our earlier papers [1, 2].

2. Related work

Refraction in computer vision has been considered in various ways: it can be considered as a bias to correct in order to use classic vision techniques, purposely used in an active system, or explicitly modeled in a refractive 3D reconstruction pipeline.

Refraction compensation in classic vision. Light rays from a point source naturally tend to diverge. By refraction, a lens allows to deviate their trajectories to make them converge in an *image point*. Refraction is therefore the key to optical instruments, with the caveat that manufacturing a photographic lens assembly requires very precise alignment of the lenses that make it up, in order to limit undesired effects, called *aberrations*. However, if the scene itself contains transparent, i.e. refractive, objects, this modifies the appearance of opaque objects, which appear distorted. A number of papers have addressed the correction of these distortions when the transparent object, such as a window pane, is attached to the front of the camera, in which case the distortions can be calibrated, and standard 3D reconstruction pipelines can then be employed. For example, while it is relevant to use photogrammetry to study convection in a water-filled tank, Maas shows in [3] how to take into account refraction through both sides of the glass to improve measurement accuracy. In [4], Łuczyński et al. show how to correct the images acquired by a pair of calibrated underwater cameras, protected by a glass plate, to restore epipolar geometry. The same idea of pre-correcting the images has also been explored in [5, 6], as well as in [7] where correction is carried out by neural networks. It has also been shown in [8, 9] that light field cameras can provide useful clues for correcting refraction in the images.

Active refraction techniques. On the other hand, several studies, which can be grouped under the term *active refraction*, take advantage of refraction to allow 3D reconstruction from a single view, thanks to image duplication, using either a bi-prism [10, 11], or a glass plate rotating around the optical axis [12, 13].

Estimation of a refractive interface. In many cases, the surface of a transparent object (the refractive interface) may be needed. In the case of water, Morris uses a pattern refracted through water surface in [14], Morris and Kutulakos propose to map points of interest seen through transparency [15]. Another approach is that of Ben-Ezra and Nayar [16], who fit the parameters of a surface model in order to reproduce the distorted image of an object of known geometry. The 3D reconstruction of transparent objects has

also been revisited recently under the neural network (NN) perspective [17, 18].

Bathymetry. An obvious need for understanding refraction is remote-sensing bathymetry. Its goal is to correct depth measurements in oceans and lake or river beds, which differs somewhat from our reconstruction problem, though uses Snell's law. This has been explored in numerous works, for instance in the work of Murase et al. [19], the already mentioned papers by Agrofiotis et al. [7, 5] but also Woodget et al. in [20] or Cao et al. in [21] to cite a few.

Classical framework with refraction adaptation. Classical components of 3D vision systems have been adapted to the presence of a refractive interface: calibration, absolute and relative camera pose estimation and refractive structure-from-motion (RSfM), refractive multi-view stereo (RMVS), refractive photometric stereo (RPS), and refractive multi-view photometric stereo (RMVPS). Sturm discusses generic camera models for SfM in [22], including the refractive axial camera. Chari and Sturm extends epipolar geometry in [23] in the case of a camera separated from the 3D scene by a planar interface, using a 12×12 fundamental matrix. A pinhole/axial camera model is proposed by Łuczyński et al. in [4] and calibration for it. A fringe projection system is studied by Chen et al. in [24]. Jordt an coworkers study the case of an underwater camera and its consequences in a series of papers. In [25], geometric and radiometric calibration procedures are proposed, and pose by *structure-from-motion* in [26, 27], as do Kang et al. [28], Qiao et al. [29], Suresh et al. [30], Chadebecq et al. [31], Hu et al. [32]. Specific pose optimization problems under flat refractive interfaces are discussed in [33]. In most of these papers, validation is carried out on underwater images – a relatively recent survey of underwater 3D imaging techniques can be found in [34]. Complete reconstruction pipelines have been proposed, especially for underwater camera [26, 31]. Though the majority of refractive interfaces are planar, series of other situations have also been explored. The case of a camera inside an underwater dome is studied in [35]. More complex situations have explored when seeing aerial objects from underwater through a randomly moving surface in [36, 37]. The converse situation (air to water) is explored in [38] using a differentiable/NN framework. Many authors have studied underwater extension of photometric stereo (PS), often focusing on scattering media properties while discarding photometric stereo, or attempting to incorporate via camera calibration as in [39, 40]. Narasimhan et al. incorporate explicitly refraction in PS, among other techniques, in [41]. Fan et al. [42, 43] use it together with refractive laser triangulation. Quéau et al. propose complete models for orthographic cameras and

directional lights in [44].

Inverse rendering and novel views. Inverse/differentiable rendering techniques have started to emerge recently, such as the differentiable rasterizers [45, 46], but rasterization is not well adapted to refraction. Ray-tracing/particle tracking inverse renderers are more promising [47, 48], or [49] which presents a reconstruction of an object in a cuboid refractive medium. They do not seem very robust (yet) and are very computationally heavy. Adaptations of the neural radiance field framework [50] to refraction is proposed in [51, 52], but as NeRF does, they only deal with novel view synthesis. Building on the 3D reconstruction NeuS framework by [53], a recent article proposes an extension for an object encased in a cuboid refractive medium.

In this paper, we rather assume that the camera poses have been pre-calibrated, e.g. by structure-from-motion, and we focus on 3D reconstruction by multi-view stereo (MVS). Extensions of the PMVS algorithm (*patch-based multi-view stereo*) by Furukawa and Ponce [54] have been proposed by Kang et al. [55] and by Chang and Chen [56], but the extension process is not really detailed. Agrawal et al. have also studied MVS in a refractive medium in [57], but their work remains limited to a (multi-layer) planar interface. To the best of our knowledge, the only work considering a non-planar interface is the recent work of Yoon et al. [58]. However, their objective is different, because the curved refractive medium is placed between the observer and the object, while in our case the object lies inside the refractive medium.

Overall, there is a lack of multi-view stereo technique adapted to the case of an object immersed in a refractive medium with a non-planar shape. The goal of our work is to fill this gap.

3. From Multi-view Stereo to Refracted Multi-view Stereo

3.1. Multi-view Stereo

Multi-view stereo (MVS) is a dense 3D reconstruction technique whose principle is to maximize the photometric coherence between different images in a 3D scene. An overview of the different approaches to MVS is provided in [59]. Knowing $t+1$ images of the scene and their associated $t+1$ camera poses, we choose the image of the first pose as the *reference image*. For a 3D point visible in all the images, denoted $\mathbf{P}$, we denote by $\mathbf{p} = \pi(\mathbf{P})$ its projection in the reference image and $\mathbf{p}_j = \pi_j(\mathbf{P})$, $j \in \{1, \dots, t\}$, its

projections in the t other images, called *control images*. The Lambertian assumption is written:

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

where I_j and I denote the gray level functions of the j-th control image and the reference image, and where the index z of the function π_z^{-1} is required, because 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), for example by solving 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 gray levels I_j and I is performed between the neighborhoods of $\mathbf{p}_j$ and of $\mathbf{p}$, and the use of a robust estimator is recommended (see the overview presented in [59, chapter 2]).

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 calibration matrix of the camera, this transformation is written:

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

The *reprojection* in the j-th control image is also obtained by central projection, taking into account the change in camera pose. If the rigid transformation between the reference pose and the j-th pose is characterized by the rotation matrix $\mathbf{R}_j$ and the translation vector $\mathbf{t}_j$, assumed to be known, and using a function ρ such that $\rho\left([a, b, c]^\top\right) = [a/c, b/c]^\top$, this second transformation is written:

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

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

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

The argument to be minimized in (5) may be nonlinear, nondifferentiable and/or nonconvex, which makes optimization potentially difficult. For this reason, solving (5) is usually done by an exhaustive search (*brute-force*) in a predefined list of values of depth z (see Figure 2-a). This strategy, apparently simplistic, turns out to be very efficient for the 3D reconstruction of scenes with a

sufficiently textured surface [60]. As suggested in Figure 2-b, the scenario is similar, but more complex, when the 3D scene is immersed in a refractive medium.

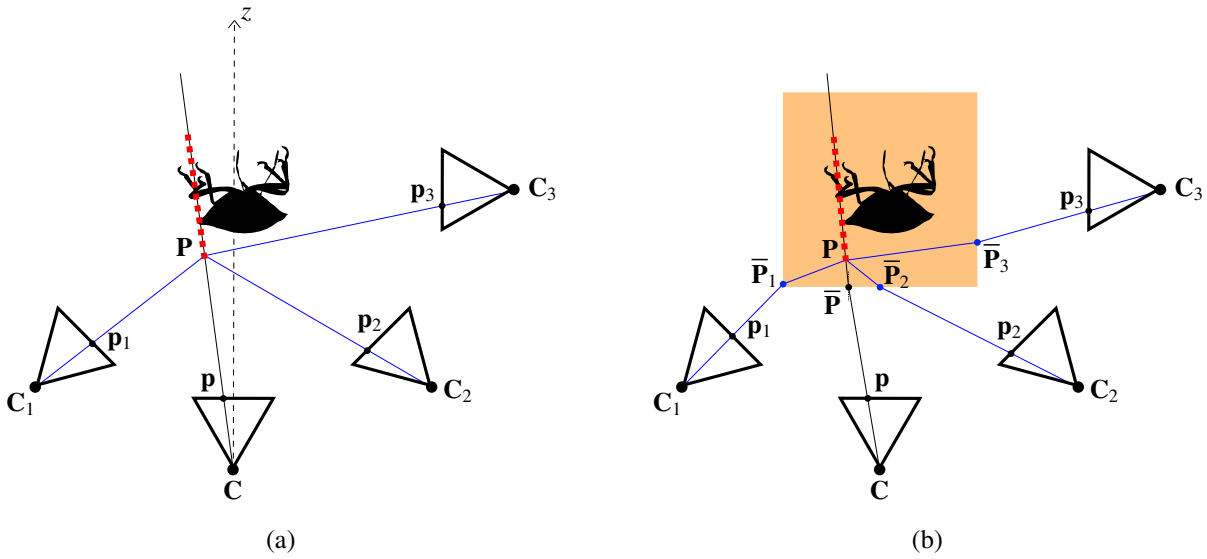


Figure 2. (a) MVS in a homogeneous medium: the different proposals for the point P, which are materialized by red dots, are reprojected in the control images. (b) MVS with refraction: the reprojection of P in the control images is more difficult to compute, due to refraction.

3.2. Refractive Multi-view Stereo

The image of an object immersed in a medium whose *index of refraction* (IoR) is strictly greater than 1, called *refractive medium*, is distorted. Among other consequences, the epipolar geometry of a pair of images is lost. Instead of corresponding to an epipolar line in the second image, a point in the first image corresponds to a curve whose shape depends on the IoR of the refractive medium and on the shape of the interface between the refractive medium and the surrounding air, called *interface*. Chari and Sturm have shown in [23] how to generalize the matrix formalism of the epipolar geometry, using a fundamental matrix of size 12×12. The epipolar geometry is mainly dedicated to the estimation of the camera pose by structure-from-motion. Since this problem has already been extended to take into account refraction [26, 31], we focus in this paper on extending the MVS technique to the case where the 3D scene is immersed in a refractive medium. This new problem, called RMVS (for *Refractive Multi-View Stereo*), consists in solving, at each point $\mathbf{p}$ of the reference image, the same problem (2) as before, with several adaptations:

- Back-projection of an image point $\mathbf{p}$ – In the presence of refraction, the back-projection of $\mathbf{p}$ requires tracing the ray coming from $\mathbf{C}$ and passing through $\mathbf{p}$, which is now a broken line (see Figure 2-b). The expression of the back-projection is thus

slightly more complicated than (3):

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

where the *point of incidence* $\overline{\mathbf{P}}$ lies at the intersection of the deprojection line of $\mathbf{p}$ with the diopter, and the unit director vector $\mathbf{v}$ of the refracted ray is obtained by the Snell-Descartes laws of refraction (see Section 3.3). In addition, $\bar{z} \geq 0$ denotes the distance between $\overline{\mathbf{P}}$ and $\mathbf{P}$ along the refracted ray (see Figure 2-b). The determination of the point of incidence $\overline{\mathbf{P}}$ can be more or less tedious, depending on the shape of the interface, while the calculation of the unitary directing vector $\mathbf{v}$ does not present any particular difficulty, as long as the normal to the interface is known with sufficient accuracy.

- Reprojection of a 3D point $\mathbf{P}$ – On the other hand, computing the reprojection $\mathbf{p}_j = \pi_j(\mathbf{P})$ in the presence of refraction is considerably more complicated than (4). Instead of a simple central projection into the j-th control image, the π_j function requires solving a *shortest optical path* problem (see Section 3.3).

- Image multiplication – As the examples in Figure 6 show, due to refraction, a single 3D point $\mathbf{P}$ can be projected to multiple points in the image. However, this does not really complicate the problem, since each of these projections can be used equivalently to solve (2).

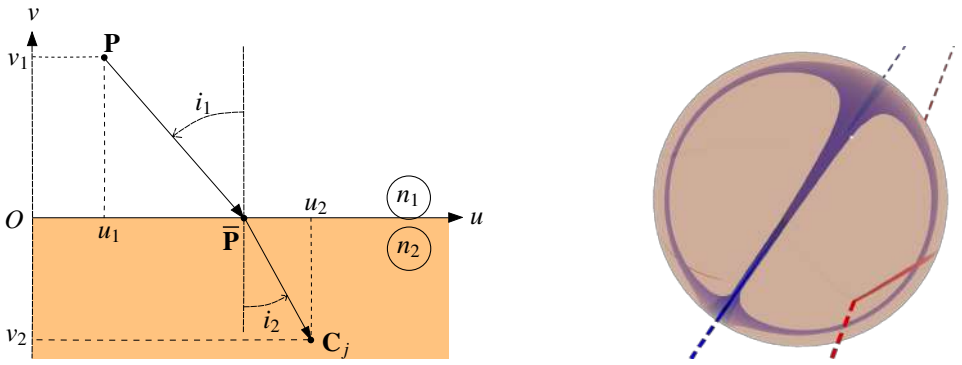
Compared to the MVS technique, the main difficulty of RMVS is the reprojection $\mathbf{p}_j = \pi_j(\mathbf{P})$, $j \in \{1, \ldots, t\}$, of a 3D point $\mathbf{P}$ into the different control images. Let us first consider the case of a plane diopter, before tackling the case of a diopter of any shape.

3.3. Planar Interface

According to the first Snell-Descartes law on refraction, the refracted ray is in the *plane of incidence*, defined by the incident ray and the normal to the dioptre in $\overline{\mathbf{P}}$, which allows us to represent the phenomenon in a plane (see Figure 3-a).

By noting i_1 the angle between the normal to the dioptre and the incident ray, in a medium of IoR n_1, and i_2 the angle between this same normal and the refracted ray, in a medium of IoR n_2, the second Snell-Descartes law on refraction is written:

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



(a) (b)

Figure 3. (a) Second Snell-Descartes law on refraction. (b) Synthetic images of two straight lines (in red and in blue) crossing a refractive sphere.

In the case of a planar interface, squaring the two members of (7), and using the notations in Figure 3-a, we have:

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

To determine the point of incidence $\overline{\mathbf{P}}$, we have 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 easy to express in terms of u_1, v_1, u_2, v_2 and $\alpha = n_2/n_1$ [57].

The image of a point $\mathbf{P}$ is unique if the interface is planar, which implies that Equation (9) has a unique real solution, which can be obtained e.g., by the Newton-Raphson method. The computation of $\mathbf{p}_j = \pi_j(\mathbf{P})$ is thus performed in two steps. First, determine the point of incidence $\overline{\mathbf{P}}$ by solving (9). Then, project this point into the j-th control image, according to (4).

3.4. Interface of Any Shape

The *Huygens-Fresnel principle* allows to predict the wave surfaces of any luminous flux, which are orthogonal to the light rays. A discrete version of the calculation of wave surfaces is provided by Dijkstra's algorithm [61], which allows to find the shortest path between two vertices of a graph. To compute the path of the light rays, it is therefore possible to divide the scene into voxels, which constitute the vertices of an undirected graph. However, the problem is greatly simplified if we assume that the refractive medium is homogeneous.

Indeed, if the IoR is the same at any point of the refractive medium, the light propagates in straight line, as in the air. The path of a light ray between a 3D point $\mathbf{P}$ and the center of projection $\mathbf{C}_j$, $j \in \{1, \ldots, n\}$, is thus a broken line with only one break at the

interface (see Figure 2-b). As we have already said at the end of the previous section, finding the shortest optical path between $\mathbf{P}$ and $\mathbf{C}_j$ comes down to finding the point of incidence $\overline{\mathbf{P}}$. If the interface is planar, we know from Section 3.3 that this is equivalent to solving a quartic equation, but if the shape of the dioptre is more complex, the *analytical* solving is generally impossible.

It is legitimate to ask whether the π_j transformation preserves the alignment of the points, i.e. whether the π_j image of a refracted light ray, which is a straight line, is still a straight line in the j-th control image, in which case the conjugate point of a $\mathbf{p}$ point of the reference image could be searched for along this line. The example in Figure 3-b shows that this is not the case.

To find the point of incidence $\overline{\mathbf{P}}$ in the general case, a very simple solution consists in discretizing the interface, and to minimize the optical path of the ray $(\mathbf{C}_j, \widehat{\mathbf{P}}, \mathbf{P})$ passing through one of the $\widehat{\mathbf{P}}$ points of the discretized interface, which amounts to solving a potentially complicated optimization problem by an exhaustive search in a list of "eligible" points $\widehat{\mathbf{P}}$:

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

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

In practice, the interface is discretized in the form of a 3D mesh with triangular faces, the points $\widehat{\mathbf{P}}$ eligible in the search for the point of incidence $\overline{\mathbf{P}}$ being the barycenters of the triangles visible from the center of projection $\mathbf{C}_j$ (see Figure 4). The triangle colored in blue on Figure 5 corresponds to the solution of Problem (10). To refine the result, the point $\overline{\mathbf{P}}$ is then searched in the plane defined by this triangle, according to the method described in Section 3.3. This solution is retained if it is inside the triangle. It is rejected if it is not, and a search, identical in all points, is performed on the set of adjacent triangles, which are indicated in purple on Figure 5.

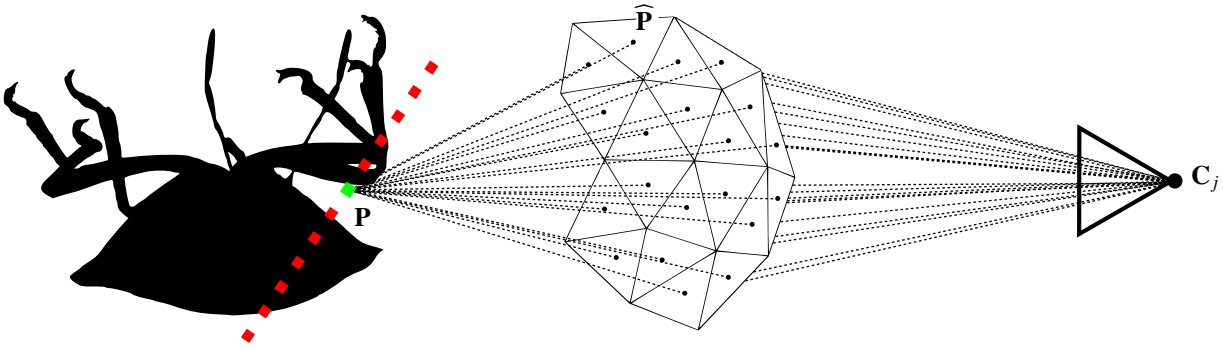


Figure 4. The point of incidence $\overline{\mathbf{P}}$ between a 3D point and the center of projection $\mathbf{C}_j$ of the j-th control camera is determined by testing the set of barycenters $\widehat{\mathbf{P}}$ of the triangles of the 3D mesh of the interface that are seen by this camera.

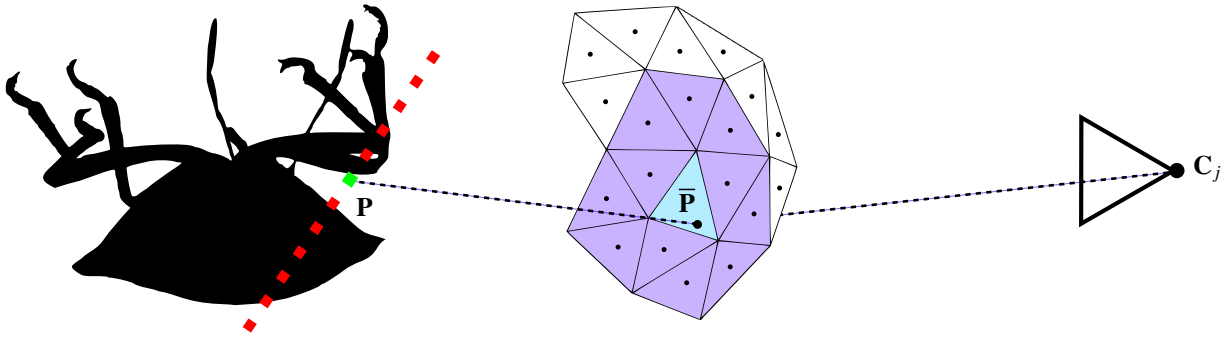


Figure 5. Once the triangle corresponding to the solution of Problem (10) has been identified (triangle indicated in blue), the search for the point of incidence $\overline{\mathbf{P}}$ is refined using the method described in Section 3.3. In the case where the solution of this second problem is outside 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 start by validating the proposed method on a scene consisting of an insect, called *graphosoma*, with a size of about 30 *mm*, immersed in a refractive cube whose IoR is equal to that of the epoxy resin ($n_2 = 1.56$). The focal length of the camera is equal to 50 *mm*, and the average distance from the scene to the camera is about 180 *mm*. Figure 6 shows two synthetic images (among 18) of this scene, computed with the ray tracing functions of the software *Blender*.

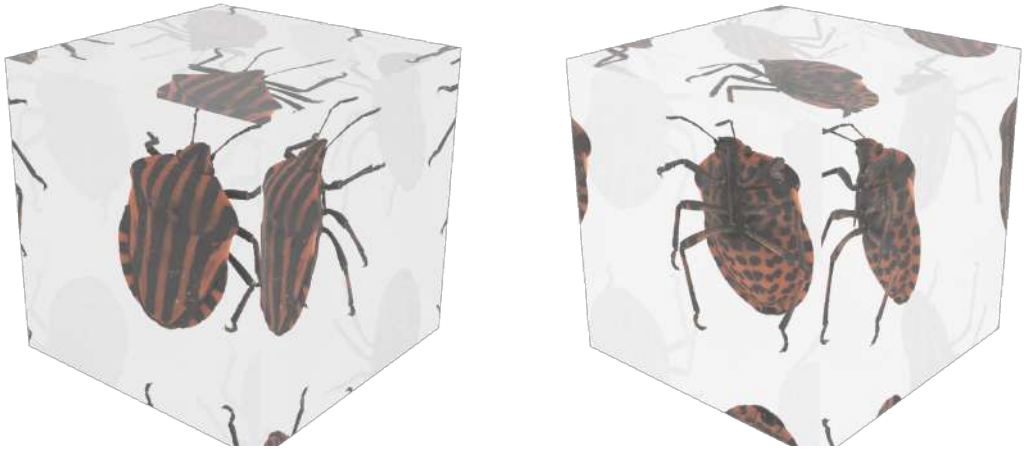


Figure 6. Two synthetic images (among 18) of a graphosoma immersed in a cube of epoxy resin. In both cases, the insect is visible through three faces of the cube (only partially, regarding the top face). The other fragments of images of the insect visible near the edge of the cube silhouette, which are due to reflection phenomena, are not used by our solving method. Source of the 3D model: Digital Archive of Natural History [62].

Figure 7 shows three views of the colored 3D point cloud reconstructed by our RMVS solving method. Figure 8 shows the same result, after "cleaning" the 3D point cloud by the *Connected-component labeling* tool of 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 8. Result of Figure 7 after "cleaning" the 3D point cloud by the *Connected-component labeling* tool of the *Cloud Compare* software.

As a comparison, Figure 9 shows the result obtained by MVS from the same data, i.e. without taking refraction into account: because of the demultiplication of the image of the graphosoma, it is badly reconstructed, and three copies even appear. On the other hand, this improvement is obtained at the cost of a much higher computation time: we go from 16 minutes for the result of Figure 9, to 24 hours for the one of Figure 7 (CPU Intel Xeon Silver 4110 2.10 GHz using all 32 threads on parallel computing), for a number of 3D points of the order of 500.000 in both cases. Note however that the computation time could be greatly reduced if, among the barycenters $\widehat{\mathbf{P}}$ of the mesh triangles (see Figure 4), only those that project inside the insect silhouette, in the considered control image, were tested.

This first example allows us to detail the solving method. The 3D reconstruction of Figure 7 results from the merging of eight colored 3D point clouds, each of these clouds being obtained as follows:

Figure 9. 3D reconstruction obtained by MVS, without taking refraction into account: because of the demultiplication of the image, the graphosoma is badly reconstructed, and it even appears copied three times.

- One of the images is chosen as reference image. Five other images are chosen as control images: for four of them, the "main" image of the insect is seen through the same face of the cube as in the reference image; for the fifth control image, it is an adjacent face.

- For each pixel **p** of the reference image, for each point **P** located on the refracted ray resulting from the deprojection of **p**, for each control image, and for each face of the cube visible in the control image (there are at most three faces visible per control image), a quartic equation of type (9) is solved by the Newton-Raphson method.

- Once the solution has been found, if its projection $\mathbf{p}_j$ in the j-th control image is located inside the silhouette of the insect in the face concerned, then the similarity between the neighborhoods of **p** and $\mathbf{p}_j$ is calculated, using a sufficiently robust estimator, in this case the sum of absolute deviations (SAD).

- In the case where the SAD can be calculated for several faces of the j-th control image (at most three), only the smallest value is retained.

- Otherwise, that is to say if no SAD can be calculated, another point **P** is tested, among a predefined list of 3D points.

- The retained point **P** is the one which minimizes the SAD. Its color is that of the pixel **p** of the reference image.

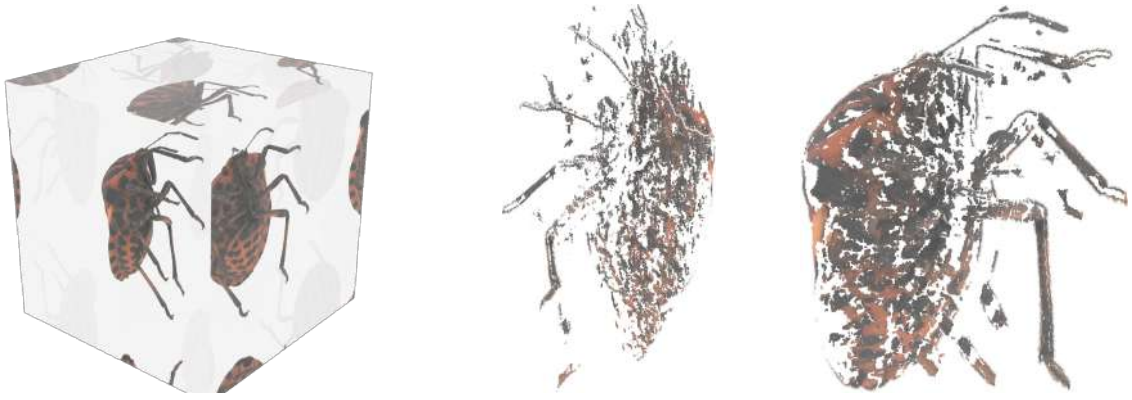
- If no SAD can be calculated, no 3D point is associated with pixel **p**.

Given that all the images of the graphosoma are synthetic, it is possible to measure the deviations from the ground truth of each of the eight 3D point clouds whose fusion produces the result of Figure 7. The square root of the mean and the square root of the median of the squared deviations are reported in Table 1. These values are very low, compared to the scale of the reference 3D model and the distance from the 3D model to the camera, which allows us to quantitatively validate 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 1. 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 shows the average of these estimates across the eight 3D point clouds.

Figure 10-a shows that the image of a point **P** located inside the refractive medium can be multiplied. These multiple images correspond to as many local minima of the optical path between a 3D point **P** and the center of projection (Fermat's principle). It is therefore possible to match these different images of the insect, which amounts to applying our RMVS solving method using only one view, as was done in [10, 11]. The result (see Figures 10-b and 10-c) is imprecise and incomplete, since it consists of a single point cloud, but the technique differs from other single-view 3D reconstruction techniques, such as shape-from-shading [63], because it is based on the principle of triangulation.

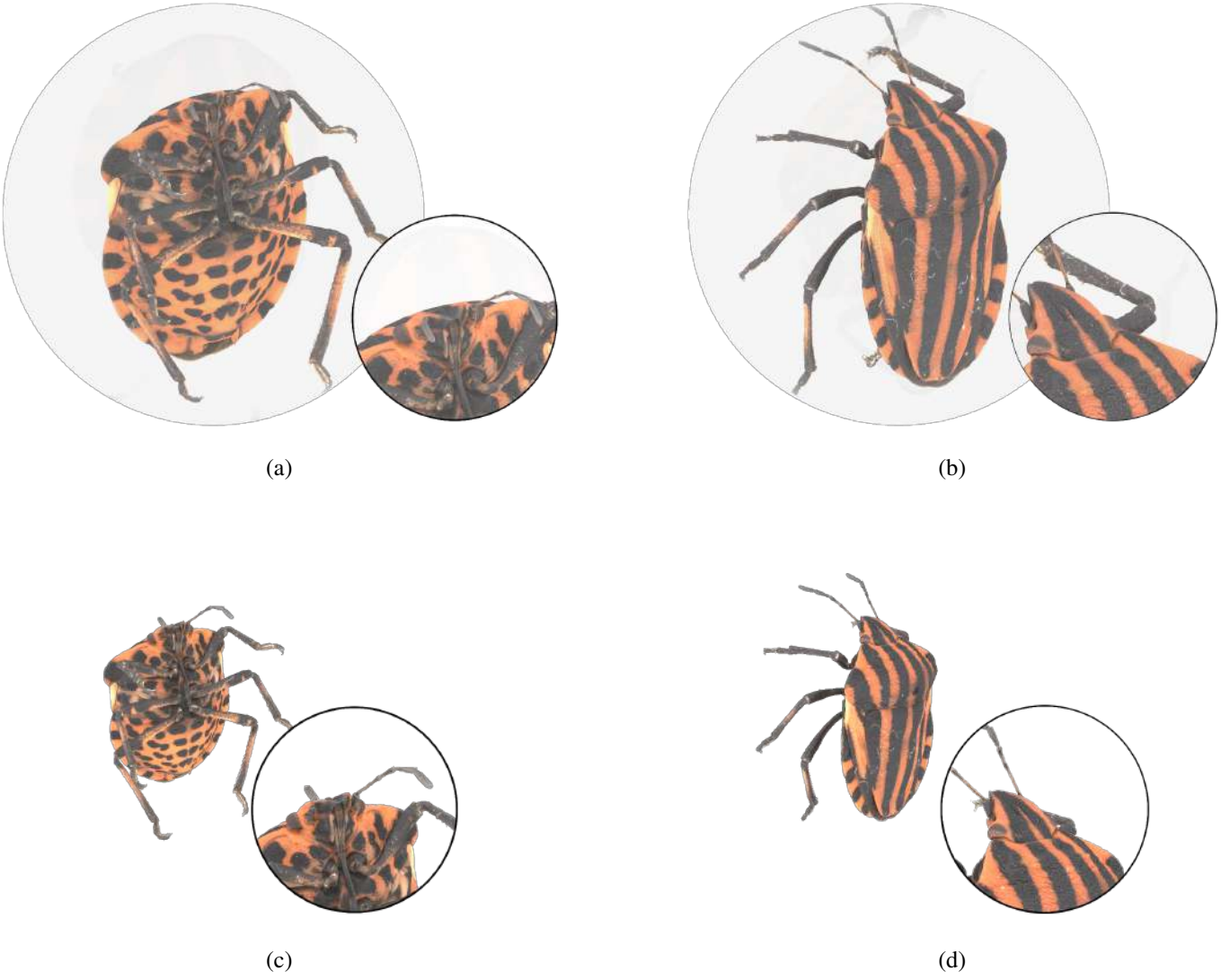


(a) (b) (c)

Figure 10. (a) Example where the image of the graphosoma is detripled. A 3D point cloud can be calculated from this single image, by choosing the "main" image (right face) as reference image. (b-c) Two views of the 3D point cloud reconstructed by our RMVS solvind method, from this single image.

4.2. Spherical Interface

The second test is carried out on a graphosoma immersed in a sphere of epoxy resin. Figure 11 shows two synthetic images of this new scene, and puts them next to images of the graphosoma seen from the same two angles, but outside the refractive medium. Besides the magnifying effect due to the convex shape of the block of epoxy resin, which acts as a convex lens, the main difference between these two pairs of images lies in the legs and insect antennae, which seem deformed by refraction. It is noteworthy that the images in Figures 11-a and 11-b are not multiplied, unlike the images in Figure 7. They are calculated by ray tracing, approaching the sphere by a triangular mesh comprising 327,000 faces.



(a) (b)

(c) (d)

Figure 11. (a-b) Two synthetic images of the graphosoma immersed in a sphere of epoxy resin. (c-d) Synthetic images of the graphosoma seen from the same two angles, but outside the refractive medium. In addition to the magnifying effect due to the convex shape of the block of epoxy resin, legs and antennae appear deformed.

Figure 12 shows from three angles the colored 3D point cloud obtained by our RMVS solving method, from 18 images such as those in Figures 11-a and 11-b, by merging eight 3D point clouds. Not only do the legs and antennae of the insect from these different clouds coincide, but very fine details are also reconstructed. At this point, it is worth remembering that the 3D point clouds are merged with no post-processing, other than the cleaning by the *Connected-component tool labeling* of the *Cloud Compare* software. On the other hand, the very high number of faces of the sphere causes a strong increase in the computing 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 a sphere of epoxy resin, seen from three angles, obtained by our RMVS solving method, from 18 images such as those in Figures 11-a and 11-b, after cleaning by the *Connected-component labeling* tool of the *Cloud Compare* software.

For comparison, Figure 13 shows that, without taking refraction into account, MVS fails to reconstruct the 3D shape, even roughly. Ghosted legs and antennae appear, which reflects the lack of consistency between the eight 3D point clouds.

Figure 13. 3D reconstruction by MVS, from 18 images such as those in Figures 11-a and 11-b: due to the lack of consistency between the eight 3D point clouds, ghost legs and antennae appear.

The interface discretization scale is guided by the need to find a compromise between precision and computing time. Table 2 shows the influence of a decrease in the number of triangular faces of the 3D mesh of the sphere, which means a less precise knowledge of the interface, through the two estimators already used in Section 4.1 (RMSE and RMedSE), as well as the percentage of 3D points actually reconstructed, and the CPU time.

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
Percentage of 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 2. Influence of a decrease in the number of triangular faces of the 3D mesh of the sphere on the 3D reconstruction of the graphosoma by our RMVS solving method (the numbers of faces are indicated in parentheses).

Figure 14 shows the 3D reconstructions corresponding to Table 2. As expected, the 3D points which "disappear" first, i.e., those conjugated with pixels p for which no SAD similarity value can be calculated, are located on the thinnest parts of the 3D model, namely the legs and the antennae.

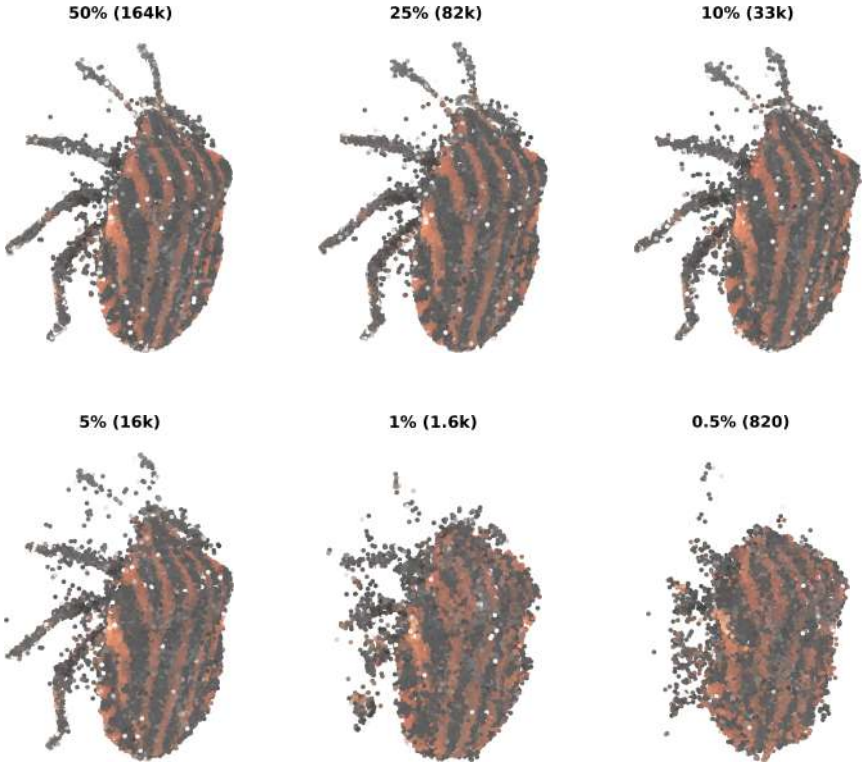


Figure 14. Six 3D reconstructions of the graphosoma immersed in a sphere of epoxy resin, when the percentage of triangular faces of the sphere used by our RMVS solving method decreases (see Table 2). Figure 12 shows the result when all the 327,000 faces are used.

4.3. Other Interfaces

Figure 15 shows two synthetic images of the graphosoma immersed in a regular dodecahedron made of the same epoxy resin. The 3D reconstruction obtained by our RMVS solving method is illustrated in Figure 16. The deviations from the ground truth are hardly higher than those of Table 1, even though the images used are much more difficult to interpret, in terms of 3D, than those of Figure 7: if the RMSE, computed on the whole point cloud, goes from 0.57 *mm* to 1.10 *mm*, the RMedSE increases much less, from 0.30 *mm* to 0.35 *mm*. This increased value of the RMSE probably comes from the significant deformation of the images, so that the similarity measurement by the SAD estimator is somehow biased.

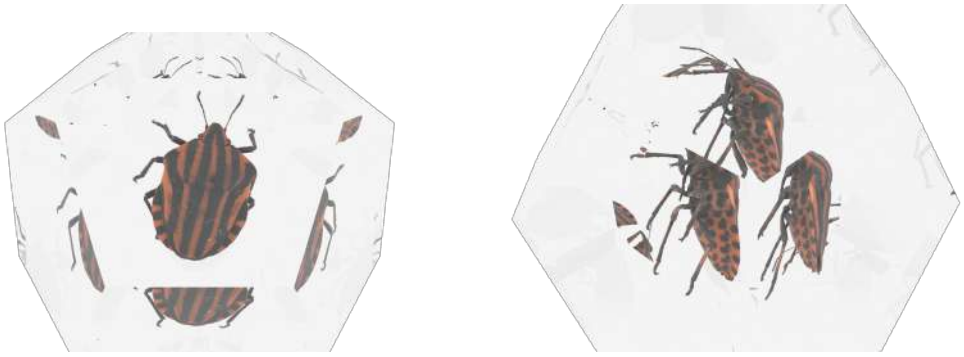


Figure 15. Two images of the graphosoma immersed in a regular dodecahedron of epoxy resin.

Figure 16. 3D reconstruction of the graphosoma immersed in a regular dodecahedron of epoxy resin, seen from three angles, obtained by our RMVS solving method from 18 images such as those in Figure 15, after cleaning with the *Connected-component labeling* tool of the *Cloud Compare* software.

Figure 17 shows that images can be distorted even more with a block of any convex shape. The 3D reconstruction obtained by our RMVS solving method (see Figure 18), which remains faithful to the original, is still less precise than that of Figure 7, because for some grazing rays (see Figure 3-a), the angle i_2 of the Snell-Descartes law (7) is very close to $\pi/2$. Hence, as the derivative of function arcsin tends to infinity at 1, this leads to inaccuracies in the calculation of the angle i_1 of Equation (7). By comparison,

Figure 19 shows that, without taking refraction into account, the result really looks like a random 3D point cloud.

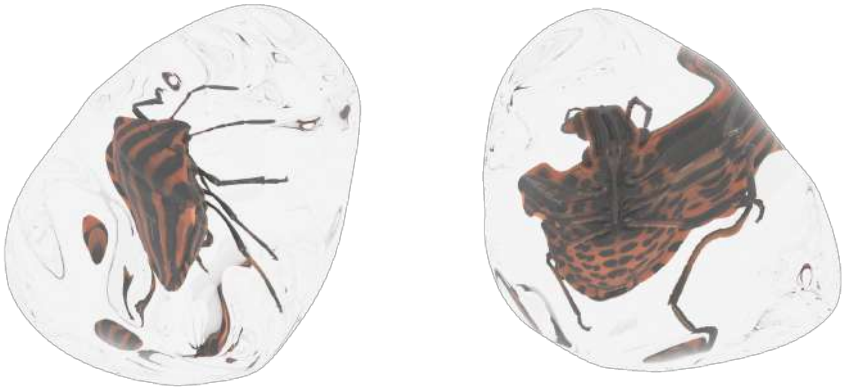


Figure 17. Two images of the graphosoma immersed in a convex block of epoxy resin.

Figure 18. 3D reconstruction of the graphosoma immersed in a convex block of epoxy resin, seen from three angles, obtained by 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 looks like 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 2), the poses of the camera and the IoR of the transparent medium.

5. Implementation on Real Images

The main difficulty in implementing our 3D reconstruction method on real images is estimating the camera poses. We could use the *refractive structure-from-motion* method proposed in [31], but this would require knowing the IoR of the medium, which is one of the unknowns. As we also need to recover the interface 3D shape, we propose in this section a method of simultaneously estimating the poses of the camera and the interface 3D shape. The method combines *shape-from-silhouette* [64] with point correspondences across the views and does not involve the IoR. The latter can therefore be estimated a posteriori, as we will see in Section 5.2.

5.1. Estimating the Camera Poses and the Interface 3D Shape

In this subsection, we describe a method for collecting the camera poses and the 3D shape of the interface, expressed in a common 3D frame, consisting of fixing the camera in front of the object to be analyzed, positioned on a turning table. The rotation of the table acts as if the camera was rotating around the object. The simultaneous estimation of the poses of the camera and the interface 3D shape is performed by shape-from-silhouettes.

This technique has the advantage of not being dependent on the IoR of the refractive medium. Nevertheless, it is hopeless to solve the problem for an object of any shape. Indeed, the principle of shape-from-silhouettes is to compute a volume enclosing the object by intersecting the deprojections of its silhouettes. However, to hope for an accuracy that meets our needs, it would be necessary to have an infinite number of poses, unless we restrict ourselves to the particular case of a polyhedral interface, whose shape can be characterized by a small number of vertices. In the case of insects trapped in amber, this is quite realistic since it is possible, as already said in Section 1, to cut the amber in order to give it a polyhedron shape i.e., a a surface composed of several planar faces.

Now, we consider a scene consisting of a convex polyhedron with N vertices, placed on a turntable. The vertices coordinates $\mathbf{X}_n$, $n \in \{1, \ldots, N\}$, are expressed in a 3D frame $\mathcal{R}_{\text{ref}}$ attached to the turntable. The origin of $\mathcal{R}_{\text{ref}}$ is located at the intersection between the supporting plane of the table and the rotation axis, and its two first axes define the supporting plane of the table, such that the third axis is a normal vector to the table directed upwards. A series of J views of this scene is acquired by a static camera

of known intrinsics. In the j-th image, $j \in \{1, \ldots, J\}$, the image $\mathbf{x}_n^j$ of vertex $\mathbf{X}_n$ satisfies the equation:

$$\mathbf{x}_n^j \sim \mathsf{P}^j \mathbf{X}_n \tag{11}$$

where the perspective projection matrix P^j related to the j-th view is written:

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

In (12), K is the calibration matrix, $(\mathsf{R}_{\text{ref}}, \mathbf{t}_{\text{ref}})$ define the change of 3D coordinates from $\mathcal{R}_{\text{ref}}$ to the camera frame, and matrix R_{θ^j} denotes the rotation of the table by an angle θ^j around its rotation axis:

$$\mathsf{R}_{\theta^j} = \begin{bmatrix} \cos\theta^j & -\sin\theta^j & 0 \\ \sin\theta^j & \cos\theta^j & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

In homogeneous Cartesian coordinates $\mathbf{x} = [x, y, 1]^\top$, an ellipse can be 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 interior (resp. exterior) of the ellipse is given by the points $\mathbf{x}$ for which $\mathbf{x}^\top \mathsf{A} \, \mathbf{x} < 0$ (resp. $\mathbf{x}^\top \mathsf{A} \, \mathbf{x} > 0$).

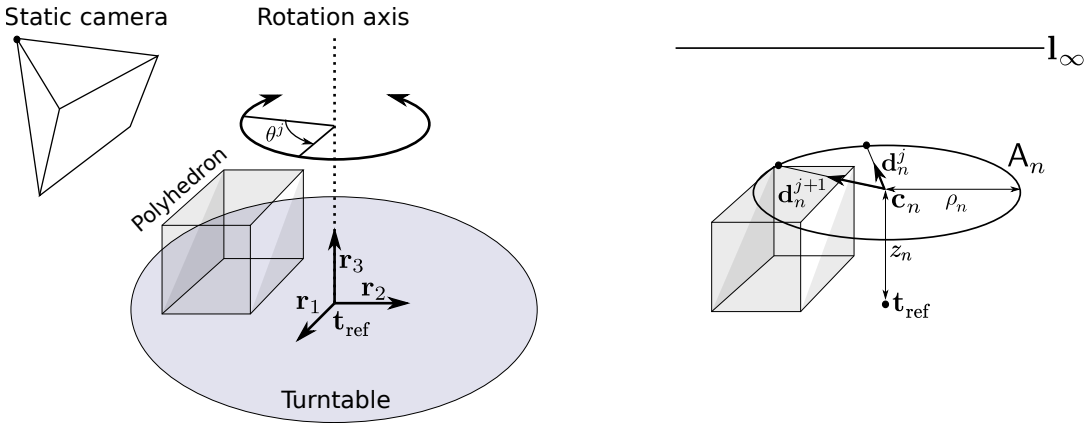


Figure 20. (a) Overview of the acquisition setup. The polyhedron is placed on a turntable. The views are acquired by a static camera. The origin $\mathbf{t}_{\text{ref}}$ is the intersection of the table with its axis of rotation. The rotation matrix R_{ref}, with columns $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$, describes the pose of the turntable with respect to the camera frame. (b) The image of a vertex, located at a distance ρ_n of the rotation axis, describes an ellipse A_n in the pixel plane. Its imaged center $\mathbf{c}_n$ satisfies the pole-polar relationship $\mathbf{c}_n = \mathsf{A}_n^{-1} \mathbf{l}_\infty$.

A first step consists in creating polygonal silhouettes of each view. This is obtained by simple operations: background subtraction from a reference image, thresholding, morphological processing, extraction, and simplification of the convex hull. Then the collection V of all the polygon vertices for all the silhouettes is created.

(a) (b)

Figure 21. (a) One out of 40 images of a parcel placed on a turntable. The silhouettes of the parcel, considered as a convex polyhedron, are extracted in all views. (b) All the silhouettes vertices, displayed in red.

In a second step, V must be robustly partitioned in subsets of points located on common ellipses, as these ellipses should be the images of *parallel circular trajectories* in 3D space. The partition size, which should match the number of vertices of the polyhedron, is unknown. A solution to this partitioning problem is described in [2]. The authors propose to exploit the parallelism of the vertices trajectories. The images of the turntable circular points (ICP) [65], noted $\mathbf{h}_1 \pm i\mathbf{h}_2$, are estimated along with the correspondences and are assumed to be known is the sequel. See [2] for more details.

Assuming that the ICP $\mathbf{h}_1 \pm i\mathbf{h}_2$, the correspondences $\{\mathbf{x}_n^j\}$ and the calibration matrix K are known, the problem consists in estimating the positions of the vertices and the poses of the polyhedron in the camera frame. More precisely, it is to compute the rotation matrix R_{ref}, the translation vector $\mathbf{t}_{\text{ref}}$, the 3D coordinates of the vertices $\{\mathbf{X}_n\}_{n\in\{1,\dots,N\}}$ and the angles $\{\theta^j\}_{j\in\{1,\dots,J\}}$. Computing R_{ref} and $\mathbf{t}_{\text{ref}}$ is performed using the method proposed in [66].

The angle of rotation θ^j is the angle of rotation of the turntable from a reference position, for example at the time of the first acquisition, and its value in view number j. Angle θ^j is computed as:

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

where $\theta^{k,k+1}$ is the angle of rotation between two consecutive acquisitions k and $k+1$. The value of $\theta^{k,k+1}$ is obtained by computing

the median of the cosines of the estimated angles from the visible vertices:

$$\theta^{k,k+1} = \operatorname{acos}\left(\underset{l\in\mathcal{D}^k}{\operatorname{median}}\left\{\mathbf{d}_l^{k\top}\mathbf{d}_l^{k+1}\right\}\right) \tag{15}$$

where $\mathcal{D}^k \subset \{1,\ldots,N\}$ denotes the set of vertex indices that have been detected in the k and $k+1$ images, and where $\mathbf{d}_l^k$ and $\mathbf{d}_l^{k+1}$ denote the unit vectors pointing towards the images by H^{-1} of the corresponding points $\mathbf{x}_l^k$ and $\mathbf{x}_l^{k+1}$ in the consecutive views k and $k+1$, that is:

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

and likewise for $\mathbf{d}_l^{k+1}$. In (16), function f is such that $f([u,v,w]^\top) = [u/w, v/w]^\top$, and $\mathbf{c}_l$ is the homogeneous coordinate vector of the image of the centre of the trajectory, supposedly circular, of the vertex number l, obtained from the pole-polar relationship $\mathbf{c}_l = \mathsf{A}_l^{-1}\mathbf{l}_\infty$, where $\mathbf{l}_\infty$ denotes the vanishing line vector of the table plane given by the cross-product $\mathbf{l}_\infty = \mathbf{h}_1 \times \mathbf{h}_2$, and A_l the matrix of the ellipse image of the trajectory of the vertex number l. The table is assumed to rotate in the trigonometric direction during the acquisition. The values of $\theta^{k,k+1}$ are supposed to be between 0 and 180 degrees.

At this stage, all the camera poses are known. The 3D coordinates of the vertices $\{\mathbf{X}_n\}_{n\in\{1,\ldots,N\}}$ are obtained by triangulating the correspondences $\{\mathbf{x}_n^j\}$. Camera poses and vertices coordinates are then refined through a bundle adjustment minimizing the Euclidean distances between the correspondences and their reprojections.

5.2. Validation on Real Images

The estimation of the index of refraction (IoR) can be performed using a dedicated instrument called refractometer, but we propose in this subsection an estimation method that takes advantage of the joint estimation of the camera poses and of the interface 3D shape, described in the previous subsection. Indeed, our RMVS solving method, which is described in Section 3 and validated on synthetic images in Section 4, can be tested for different values of the IoR, for example in a range around its "plausible" value. It remains to find a sufficiently discriminating criterion for this estimation. Figure 22 shows that the number of effectively reconstructed 3D points constitutes such a criterion.

We are now able to test the complete RMVS resolution pipeline on real data, on the sole condition that the interface is polyhedral. Figures 23 and 24 show two objects on which tests were carried out. These are two parallelepipeds made of resin containing,

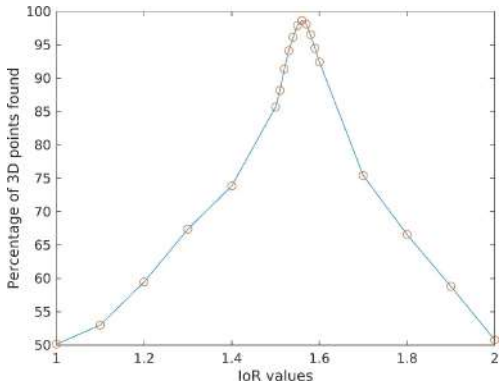


Figure 22. Evolution of the percentage of reconstructed 3D points in function of the index of refraction of the medium in which the graphosoma is immersed. The IoR used when simulating the images ($n_2 = 1.56$) corresponds precisely to the maximum of this curve, which validates the proposed criterion.

respectively, a beetle and a grasshopper. Figure 25 shows how the previous method, applied to the images of the beetle, gives an estimate of the resin IoR equal to $n_2 = 1.50$.

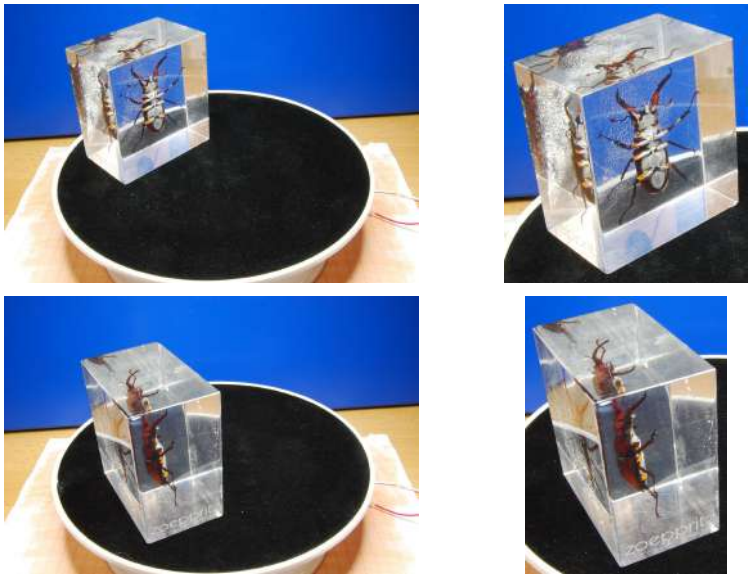


Figure 23. Left: two real images of a beetle immersed in a parallelepipedic block of resin, placed on a turntable. Right: zooms on the block.

The 3D reconstructions of these two insects are presented in Figures 26 and 27. Clearly, the beetle is better reconstructed than the grasshopper. This is mainly due to the characteristics of the resin block containing the beetle, which do not perfectly satisfy the assumptions on which our RMVS solving method is based. On the one hand, at least one of the faces is not as planar as the others. On the other hand, the resin consists of layers visible to the naked eye, which suggests that the light rays inside the refractive medium are not perfectly straight. But another important difference between the results of Figures 26 and 27 concerns the insects themselves. Indeed, some parts of the grasshopper's body appear relatively translucent, which calls into question one of

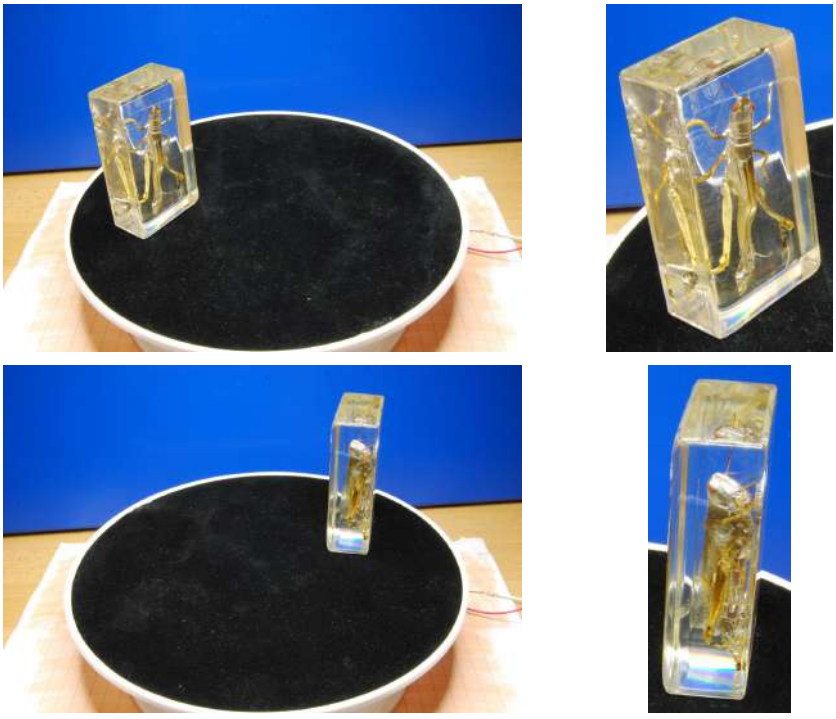


Figure 24. Left: two real images of a grasshoper immersed in a parallelepipedic block of resin, placed on a turntable. Right: zooms on the block.

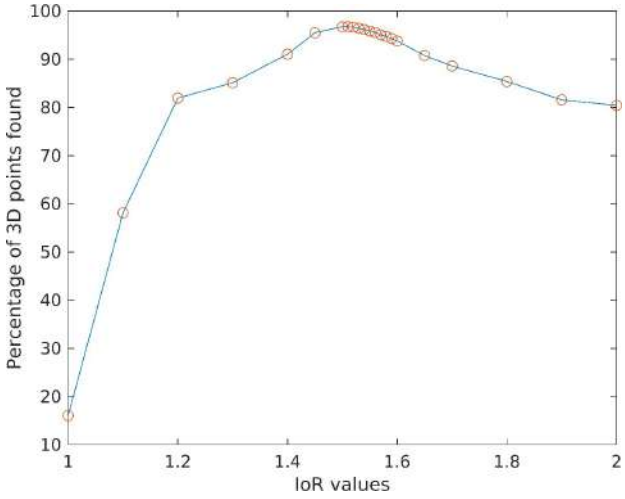


Figure 25. Evolution of the percentage of reconstructed 3D points, in function on the IoR of the medium in which the graphosoma is immersed. The maximum of this curve gives us an estimate of the IoR equal to $n_2 = 1.56$.

the basic assumptions of the MVS technique and its variants, namely that the surface to be reconstructed is assumed to be opaque and Lambertian.

A final test carried out on the images of the beetle aims to verify, at least qualitatively, to what extent the use of an erroneous IoR value influences the result. Figure 28 shows two 3D reconstructions of the beetle, obtained with two different values of the IoR: the result on the right, obtained with a slightly overvalued IoR ($n_2' = 1.56$), is clearly less precise than the one on the left, where the

Figure 26. 3D reconstruction of the beetle from 24 images such as those in Figure 23, using our RMVS solving method.

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

value of the IoR ($n_2 = 1.50$) was estimated using the method described above (see Figure 22).

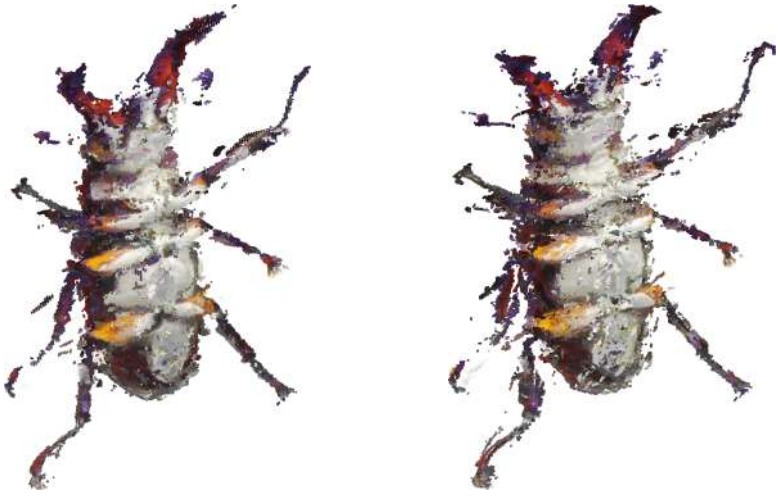


Figure 28. Comparison of our RMVS solving method, tested on the same 24 real images of the beetle (see Figure 23), using either the value of the IoR estimated by the method illustrated in Figure 22 ($n_2 = 1.50$), or a slightly overvalued IoR ($n_2' = 1.56$). The first result is obviously more accurate.

6. Conclusion and Perspectives

In this paper, we have shown how to adapt the multi-view stereo technique to the case where the object of interest is immersed in a refractive medium. Knowing that refraction distorts the images, it is necessary to take this phenomenon into account to model the path of the light rays. More specifically, we have proposed a fully discrete RMVS solving method. Our first results on real data are encouraging, although there are still many obstacles to overcome before this study leads to an operational method that can be used by entomologists.

As has already been said, the relatively limited quality of the result of Figure 27 comes from both the block of resin and the object it contains, neither of which fully satisfies the assumptions on which our RMVS solving method is based. A first perspective is therefore to make the entire pipeline more robust to the most predictable defaults. In particular, the problem becomes much more complex if the index of refraction of the medium is not homogeneous. Moreover, even under the Lambertian assumption, if the refractive medium is colored, the appearance of a 3D point can vary from one image to another, depending on the distance traveled by the light in the medium. Finally, focus blur, which is a phenomenon difficult to avoid at small scales, was ignored in this study. Taking all these effects into account will certainly improve the accuracy of the results.

The efficiency of our RMVS solving method could also be improved. To give an order of magnitude of the complexity of this fully discrete solving strategy, each of the eight 3D point clouds which give, by fusion, the result of Figure 7, is obtained with $t = 5$ control images. The reference image has approximately 1.5×10^5 pixels. The block of epoxy resin being discretized in 200 slices, the complete resolution of the problem (2) requires to solve approximately 10^8 equations of degree 4, which requires around 24 hours of CPU time.

References

[1] M. Cassidy, J. Mélou, Y. Quéau, F. Lauze, J.-D. Durou, Refractive Multi-view Stereo, in: Proceedings of the International Conference on 3D Vision, 2020.

[2] B. Brument, L. Calvet, R. Bruneau, J. Mélou, S. Gasparini, Y. Quéau, F. Lauze, J.-D. Durou, 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, A. Birk, Image rectification with the pinax camera model in underwater stereo systems with verged cameras, in: OCEANS 2017, 2017, pp. 1–7.

[5] P. Agrafiotis, K. Karantzalos, A. Georgopoulos, D. Skarlatos, Correcting image refraction: Towards accurate aerial image-based bathymetry mapping in shallow waters, Remote Sensing 12 (2) (2020) 322.

[6] X. Wu, X. Tang, Accurate binocular stereo underwater measurement method, International Journal of Advanced Robotic Systems 16 (5).

[7] P. Agrafiotis, D. Skarlatos, A. Georgopoulos, K. Karantzalos, 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) (2019) 2225.

[8] K. Ichimaru, H. Kawasaki, Underwater Stereo Using Refraction-Free Image Synthesized From Light Field Camera, in: Proceedings of the IEEE International Conference on Image Processing, 2019, pp. 1039–1043.

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

[10] D. H. Lee, I.-S. Kweon, R. Cipolla, A biprism-stereo camera system, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 1999, p. 87.

[11] A. Yamashita, Y. Shirane, 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, 2010, pp. 3652–3657.

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

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

[14] N. J. W. Morris, Image-based water surface reconstruction with refractive stereo, Master's thesis (2004).

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

[16] M. Ben-Ezra, S. K. Nayar, What does motion reveal about transparency?, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2, 2003, pp. 1025–1032.

[17] Z. Li, Y.-Y. Yeh, 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, 2020, pp. 1262–1271.

[18] M. Shao, C. Xia, D. Duan, X. Wang, Polarimetric Inverse Rendering for Transparent Shapes Reconstruction, arXiv preprint arXiv:2208.11836.

[19] T. Murase, M. Tanaka, T. Tani, Y. Miyashita, N. Ohkawa, S. Ishiguro, Y. Suzuki, H. Kayanne, H. Yamano, A Photogrammetric Correction Procedure for Light Refraction Effects at a Two-Medium Boundary, Photogrammetric Engineering and Remote Sensing 9 (8) (2008) 1129–1136.

[20] A. S. Woodget, J. T. Dietrich, 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) (2019) 2415.

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

[22] P. Sturm, Multi-view geometry for general camera models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 206–212.

[23] V. Chari, P. Sturm, Multi-View Geometry of the Refractive Plane, in: Proceedings of the British Machine Vision Conference, 2009, pp. 1–11.

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

[25] A. Jordt-Sedlazeck, R. Koch, Refractive calibration of underwater cameras, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 846–859.

[26] A. Jordt, K. Köser, R. Koch, Refractive 3D reconstruction on underwater images, Methods in Oceanography 15–16 (2016) 90–113.

[27] A. Jordt-Sedlazeck, D. Jung, R. Koch, Refractive Plane Sweep for Underwater Images, in: Proceedings of the German Conference on Pattern Recognition, 2013, pp. 333–342.

[28] L. Kang, L. Wu, Y. Wei, S. Lao, Y.-H. Yang, Two-view underwater 3D reconstruction for cameras with unknown poses under flat refractive interfaces, Pattern Recognition 69 (2017) 251–269.

[29] X. Qiao, Y. Ji, A. Yamashita, H. Asama, Structure from Motion of Underwater Scenes Considering Image Degradation and Refraction, IFAC-PapersOnLine 52 (22) (2019) 78–82.

[30] S. Suresh, E. Westman, M. Kaess, Through-water stereo slam with refraction correction for AUV localization, IEEE Robotics and Automation Letters 4 (2) (2019) 692–699.

[31] F. Chadebecq, F. Vasconcelos, R. Lacher, E. Maneas, A. Desjardins, S. Ourselin, T. Vercauteren, D. Stoyanov, Refractive Two-View Reconstruction for Underwater 3D Vision, International Journal of Computer Vision 128 (5) (2020) 1101–1117.

[32] X. Hua, F. Lauze, K. S. Pedersen, Refractive Pose Refinement, International Journal of Computer Vision (2023) 1–29.

[33] S. Haner, K. Åström, Absolute pose for cameras under flat refractive interfaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1428–1436.

[34] M. Castillón, A. Palomer, J. Forest, P. Ridao, State of the Art of Underwater Active Optical 3D Scanners, Sensors 19 (23) (2019) 5161.

[35] M. She, D. Nakath, Y. Song, K. Köser, Refractive geometry for underwater domes, ISPRS Journal of Photogrammetry and Remote Sensing 183 (2022) 525–540.

[36] M. Alterman, Y. Y. Schechner, 3D in natural random refractive distortions, in: Three-Dimensional Imaging, Visualization, and Display 2016, Vol. 9867, SPIE, 2016, pp. 64–75.

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

[38] J. Xiong, W. Heidrich, In-the-wild single camera 3D reconstruction through moving water surfaces, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12558–12567.

[39] C. Tsiotsios, M. E. Angelopoulou, T.-K. Kim, A. J. Davison, Backscatter compensated photometric stereo with 3 sources, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2251–2258.

[40] Z. Murez, T. Treibitz, R. Ramamoorthi, D. J. Kriegman, Photometric Stereo in a Scattering Medium, IEEE Transaction on Pattern Analysis and Machine Intelligence 39 (9) (2017) 1880–1891.

[41] S. G. Narasimhan, S. K. Nayar, B. Sun, S. J. Koppal, Structured light in scattering media, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 1, 2005, pp. 420–427.

[42] H. Fan, L. Qi, Y. Ju, J. Dong, H. Yu, Refractive laser triangulation and photometric stereo in underwater environment, Optical Engineering 56 (11) (2017) 113101–113101.

[43] H. Fan, L. Qi, C. Chen, Y. Rao, L. Kong, J. Dong, H. Yu, Underwater optical 3-D reconstruction of photometric stereo considering light refraction and attenuation, IEEE Journal of Oceanic Engineering 47 (1) (2021) 46–58.

[44] Y. Quéau, R. Bruneau, J. Mélou, J.-D. Durou, F. Lauze, 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, 2023, pp. 691–703.

[45] S. Liu, T. Li, W. Chen, H. Li, Soft rasterizer: A differentiable renderer for image-based 3d reasoning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7708–7717.

[46] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, S. Fidler, Extracting triangular 3d models, materials, and lighting from images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8280–8290.

[47] M. Nimier-David, D. Vicini, T. Zeltner, W. Jakob, Mitsuba 2: A retargetable forward and inverse renderer, ACM Transactions on Graphics (TOG) 38 (6) (2019) 1–17.

[48] W. Jakob, S. Speierer, N. Roussel, D. Vicini, DR. JIT: a just-in-time compiler for differentiable rendering, ACM Transactions on Graphics (TOG) 41 (4) (2022) 1–19.

[49] K. Yan, C. Lassner, B. Budge, Z. Dong, S. Zhao, Efficient estimation of boundary integrals for path-space differentiable rendering, ACM Transactions on Graphics (TOG) 41 (4) (2022) 1–13.

[50] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, in: Proceedings of the European Conference on Computer Vision, 2020.

[51] M. Bemana, K. Myszkowski, J. Revall Frisvad, H.-P. Seidel, T. Ritschel, Eikonal fields for refractive novel-view synthesis, in: Proceedings of the ACM SIGGRAPH Conference, 2022, pp. 1–9.

[52] T. Fujitomi, K. Sakurada, R. Hamaguchi, H. Shishido, M. Onishi, Y. Kameda, LB-NERF: Light Bending Neural Radiance Fields for Transparent Medium, in: Proceedings of the IEEE International Conference on Image Processing, 2022, pp. 2142–2146.

[53] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, W. Wang, Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, arXiv preprint arXiv:2106.10689.

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

[55] L. Kang, L. Wu, Y.-H. Yang, Two-View Underwater Structure and Motion for Cameras under Flat Refractive Interfaces, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 303–316.

[56] Y. J. Chang, T. Chen, Multi-View 3D Reconstruction for Scenes under the Refractive Plane with Known Vertical Direction, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 351–358.

[57] A. Agrawal, S. Ramalingam, Y. Taguchi, V. Chari, A theory of multi-layer flat refractive geometry, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3346–3353.

[58] S. Yoon, T. Choi, S. Sull, Depth estimation from stereo cameras through a curved transparent medium, Pattern Recognition Letters 129 (2020) 101–107.

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

[60] M. Goesele, B. Curless, S. M. Seitz, Multi-View Stereo Revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2402–2409.

[61] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[62] Digital Archive of Natural History, `https://sketchfab.com/disc3d`.

[63] J.-D. Durou, M. Falcone, M. Sagona, Numerical Methods for Shape-from-shading: A New Survey with Benchmarks, Computer Vision and Image Understanding 109 (1) (2008) 22–43.

[64] C. E. Hernández, F. Schmitt, Silhouette and Stereo Fusion for 3D Object Modeling, Computer Vision and Image Understanding 96 (3) (2004) 367–392.

[65] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, 2004.

[66] P. F. Sturm, Algorithms for Plane-Based Pose Estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000.