A Shape-from-silhouette Method for 3D-reconstruction of a Convex Polyhedron

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

A Shape-from-silhouette Method for 3D-reconstruction of a Convex Polyhedron

Baptiste Brument[a], Lilian Calvet[a], Robin Bruneau[b], Jean Mélou[a], Simone Gasparini[a], Yvain Quéau[c], François Lauze[b], and Jean-Denis Durou[a]

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

ABSTRACT

We present a pipeline to recover precisely the geometry of a convex polyhedral object from multiple views under circular motion. It is based on the extraction of visible polyhedron vertices from silhouette images and matching across a sequence of images. Compared to standard structure-from-motion pipelines, the method is well suited to the 3D-reconstruction of low-textured and non-Lambertian materials. Experiments on synthetic and real datasets show the efficacy of the proposed framework.

1. INTRODUCTION

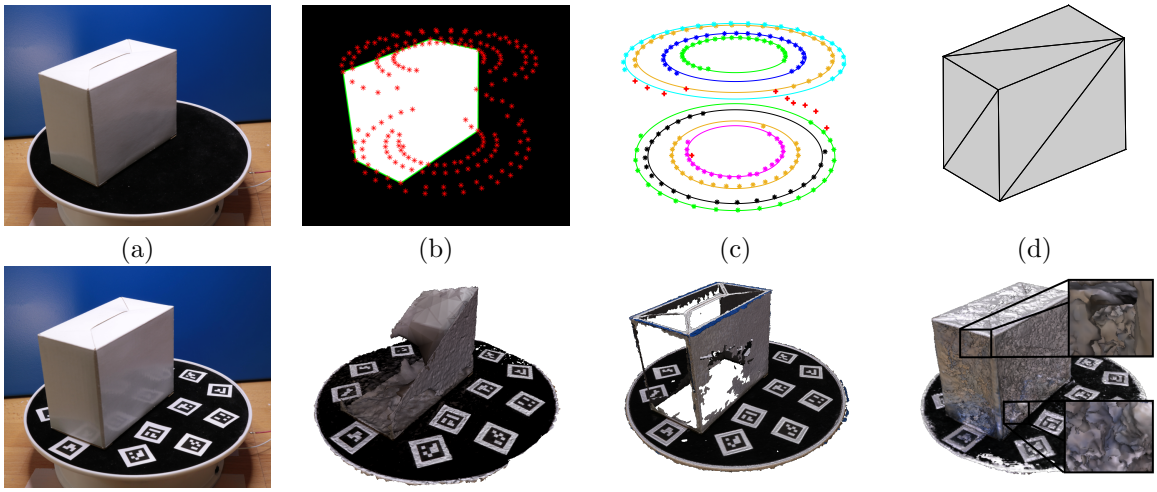
Many man-made objects have a relatively simple geometry. Among them, convex polyhedra are abundant, especially rectangular blocks. Estimating their 3D-geometry from the optical acquisition is of interest in multiple application domains. For example, nearly all parcels show polyhedric shapes (Figure 1). Automating their logistics requires knowledge of their dimensions to solve the bin packing problem. Another example is the reconstruction of artifacts encased in polyhedral transparent media (Figure 6). Knowledge of the container geometry is of utmost importance to model the distortion of the image artifact caused by refraction, so as to develop adapted 3D-reconstruction pipelines.

To estimate such a geometry, one could rely on special markers such as AruCo or April Tags (Figure 1e). Yet, this can slow down the acquisition process of large datasets and be cumbersome to manipulate. To avoid markers, we use a semi-controlled image acquisition setup: the block/object is positioned on a turntable and viewed by a static camera whose pose relative to the table is unknown. From the collection of views, we estimate the positions of the polyhedron vertices and the camera poses relative to the polyhedron by combining shape-from-silhouette[1] with point correspondences across the views (Figure 1). This allows one to 3D-reconstruct a polyhedron without assuming Lambertian materials.

The steps of our solution are: extract the silhouette of the polyhedron in each view; detect the edges of the polygon formed by each silhouette; extract the polygon vertices which correspond to the vertices of the imaged polyhedron (Figure 1b); robustly match the imaged polyhedron vertices across the views (Figure 1c); robustly 3D-reconstruct the polyhedron vertices along with its *topology* (Figure 1d). Robustness is important, especially when controlling the lighting is not possible, which may lead to some aberrant silhouette extractions. We apply the method to datasets of real turntable images of five convex polyhedra. As an evaluation of our approach, we also use the geometry and poses information to recover from images a 3D-point cloud of an insect encased in a box-shaped block of epoxy resin using the refractive MVS method.

Our main contribution is thus a robust extraction and matching algorithm for images of points whose trajectories should be parallel 3D-circles. This yields an easy-to-use, marker-free pipeline for 3D-reconstruction of convex polyhedra, which only requires a turntable. Code and datasets will be made publicly available.

After reviewing related work in Section 2, we describe our notations in Section 3. The matching of imaged polyhedron vertices is described in Section 4. An evaluation of the method is then provided in Section 5, and Section 6 draws our conclusions.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1. Top row: overview of our 3D-reconstruction pipeline. (a) One (out of 40 images) of a parcel placed on a turntable. The silhouettes vertices, displayed in red in (b), are located on ellipses (c). Our method retrieves the point correspondences from the set of points displayed in (b), namely the subsets of points belonging to the ellipses (c). From these correspondences, we perform the 3D-reconstruction of the polyhedron (d), without using markers. Bottom row: State-the-art results, where the camera poses are computed from markers (e). Multi-view stereo (MVS) results from (f) Meshroom,[2] (g) COLMAP,[3] and (h) the very recent NeRF-based instant-NGP.[4] 3D-reconstruction fails in (f-g) due to the lack of texture and glossy material, while (h) is very noisy.

2. PREVIOUS WORK

Determining the shape of an object placed on a turntable is already a well-studied problem.[5–8] Multi-view geometry can be recovered from epipolar tangents.[8] Yet, this method assumes smooth object, hence it does not work well with polyhedra. Assuming knowledge of the two-view fundamental[5] is also a restrictive assumption. It is for example not possible when reconstructing a tetrahedron comprising four vertices: four point correspondences are available while seven are required to estimate the fundamental matrix, or five in a calibrated scenario.

The geometry can also be recovered by the shape-from-silhouette technique, which computes an object envelope from its silhouettes in different views.[1,9,10] In the case of a polyhedron, reconstructing the planar faces is however impossible in practice because, for every face, the camera optical center must be located in the face supporting plane in at least one view, which is very unlikely. In addition, the camera poses are required and must be computed beforehand. A workaround could consist in placing planar markers[11,12] on the turntable, to compute camera poses before doing the 3D-reconstruction either by shape-from-silhouette or multi-view stereo (MVS). Yet, shape-from-silhouette would fail due to markers moving along with the polyhedron, making the silhouette extraction fail in most views. On the other hand, MVS[13,14] does not perform well in the case of a poorly textured object or non-Lambertian materials (Figure 1f), or for transparent objects (Figure 6e). In the latter case, MVS delivers a mesh comprising a large number of vertices and edges which do not correspond to the real polyhedron faces. Retrieving the latter requires complex post-processing: mesh denoising, holes filling, face segmentation, plane fitting, and plane intersection computation. On the contrary, our method computes the polyhedron faces directly.

The method we propose takes inspiration from the work of Jiang et al.,[6,7] who remark that the trajectories of the points on an object placed on a turntable are parallel 3D-circles whose centres are located on the rotation axis. Their images, therefore, lie on elliptical images of these trajectories. Jiang et al. make use of these ellipses to compute the camera poses. To fit the conics, the authors assume reliable point correspondences across the images collection from marked points. Yet, in our scenario such correspondences are not available and must be computed automatically. This can be particularly challenging in the case of non-Lambertian materials, for which tracking or wide baseline keypoint-based matching algorithms[15–18] perform poorly. Instead, we use the imaged vertices of the object obtained from its polygonal silhouettes.

3. NOTATIONS

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 whose origin 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. The third axis is a normal vector to the table directed upwards.

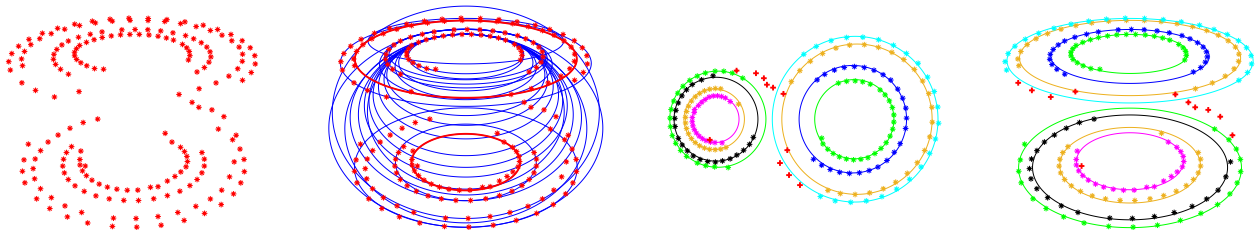
A series of J views of this scene is acquired by a static camera with known intrinsics, and we denote by $\mathbf{x}_n^j$ the image of vertex $\mathbf{X}_n$ in the j^{th} image, $j \in \{1, \ldots, J\}$.

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. $\mathbf{x}^\top \mathsf{A}\, \mathbf{x} > 0$).

4. MATCHING OF IMAGED VERTICES

The 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.

In the 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. In the remainder of this section, a *point* refers to an element of V. The cardinality of V is denoted by $|V|$.



(a) (b) (c) (d)

Figure 2. Main steps of the proposed matching. The input data is the silhouettes vertices V in all views, shown in red in (a). *(i)* The ellipses formed by the point triplets showing the highest numbers of points located in their neighborhood are kept, displayed in blue in (b). *(ii)* Robust regression of the imaged circular points from the obtained ellipses. It provides a Euclidean rectification of the points V, displayed in (c), computed from an intersection of a pair of ellipses (in red in (b)) associated with the trajectories of two polyhedron vertices. *(iii)* Collection of the correspondences resulting from the partitioning of points located on circles, displayed in (c), each colour being associated with a vertex. The matching solution is shown in (d). The red crosses are unclassified vertices.

4.1 Exhaustive Search

Five points are needed to determine an ellipse. The partitioning problem can therefore be solved by an exhaustive search for subsets of five points such that a large enough number of the remaining points are located near the ellipse passing through the five selected points. In practice, the number $\binom{|V|}{5}$ of subsets of five points is too high to perform an exhaustive search. Two assumptions enable us to drastically reduce it: we assume that the rotation axis of the turntable projects vertically into the image and that the common abscissa of the centres of the ellipses is known. The estimation of an ellipse is then reduced to the computation of three parameters, namely both semi-axes and the center ordinate, requiring only three points. If the optical axis is roughly pointing towards the rotation axis, then the abscissa of the ellipse formed by the turntable is shared with the ellipses formed by the vertices trajectories. This property makes the abscissa, common to all ellipses centers, easy to get in practice, for example by manually selecting the abscissa of the turntable extremities and considering their mean values. Deviations to our assumptions are evaluated in the supplementary material.[19]

We build the $\binom{|V|}{3}$ triplets of points that can be formed from the point set V. Triplets containing at least two points extracted in the same image are removed since a vertex can only have one image per view. The ellipses are estimated for the set of remaining triplets. The obtained set of ellipses is denoted E. For a polyhedron of general geometry in a noise-free scenario, when the number of views is greater than the number of vertices, i.e. $J > N$, the ellipses of E passing through the largest number of points are the trajectories of the images of the polyhedron vertices, and the associated subsets of points the solution to our matching problem. The computed ellipses are sorted according to the number of points located in their neighborhood. Ellipses with the highest scores should correspond to the subsets of points with the highest probability of representing images of the same vertex. This number of points, or score, is computed as follows.

Score – A point is *located near an ellipse* A if it is inside the elliptical envelope $\mathcal{E}(\mathsf{A}_{+\delta}, \mathsf{A}_{-\delta})$: $\mathsf{A}_{+\delta}$ (resp. $\mathsf{A}_{-\delta}$) denotes the ellipse having the same centre and orientation as A, and whose semi-axes are those of A enlarged (resp. shortened) by δ. This is an approximation of the δ-tubular neighborhood of A proposed in:[20] it avoids the computation of the distance of a point to an ellipse, which involves solving quartic equations and can become demanding. A point with Cartesian homogeneous coordinates $\mathbf{x}$ belongs to $\mathcal{E}(\mathsf{A}_{+\delta}, \mathsf{A}_{-\delta})$ if $(\mathbf{x}^\top \mathsf{A}_{+\delta}\mathbf{x})(\mathbf{x}^\top \mathsf{A}_{-\delta}\mathbf{x}) < 0$, and outside otherwise. The score of an ellipse A is the number of points that are contained in $\mathcal{E}(\mathsf{A}_{+\delta}, \mathsf{A}_{-\delta})$. The value of δ is computed automatically.[19]

Singularity A – The trajectory of a vertex located on the rotation axis of the table is reduced to a point. Such a configuration is a singularity for the score computation. Elliptical envelopes containing this point generally obtain very high scores while associating the images of the vertex with those of other vertices. We circumvent this problem by partitioning the points of V beforehand. Groups consisting of at least four very close points, i.e., located at a distance of less than a given threshold (0.01 in practice) from each other*, are excluded from the score computation.

4.2 Error Pruning

Highest score ellipses may include in some cases images of several vertices, because of the δ-tolerance in the score computation, the presence of measurement noise, or imaged trajectories crossing each other. An example of ellipses associated with the highest scores corresponding to aberrant groupings is shown in Figure 2a. However, ellipses representing the imaged vertices trajectories are among the ellipses of highest scores, reducing consequently the search space.

At this stage, the parallelism of the supporting plane of the vertices trajectories has not been used. This property can be exploited by using a pair of points at infinity, which are conjugate complex points included in the trajectories of all polyhedron vertices. They are the *circular points*[21] of the supporting plane of the turntable. Since the intersections of conics are invariant to any projective transformation,[21] the set of ellipses that are imaged vertices trajectories intersect the images of common conjugate complex points, referred to as ICP (for *images of circular points*). If the ICP are identified, it then becomes possible to classify the set of ellipses collected in Section 4.1 into two classes: those which belong to the family of imaged vertices trajectories, namely when intersecting the ICP, and the others.

ICP identification – In theory, the only set of ellipses in E, whose cardinality is strictly greater than two, that intersect in the same pair of complex conjugate points (the ICP) are the imaged vertices trajectories. This property allows us to compute the ICP of the supporting plane of the table as the solution of a robust regression problem of the intersection points of the ellipses E. Note that chance can produce a subset of at least three ellipses of E intersecting at the same pair of complex conjugate points other than the ICP. It is however highly unlikely that a subset of ellipses intersecting at the same pair of points other than the ICP has a number of elements greater than that of a family of imaged vertices trajectories. Only the subset of ellipses of the highest cardinality, sharing common intersections, is retained.

*After normalizing all the points so that they are located within $[-1, 1]^2$.

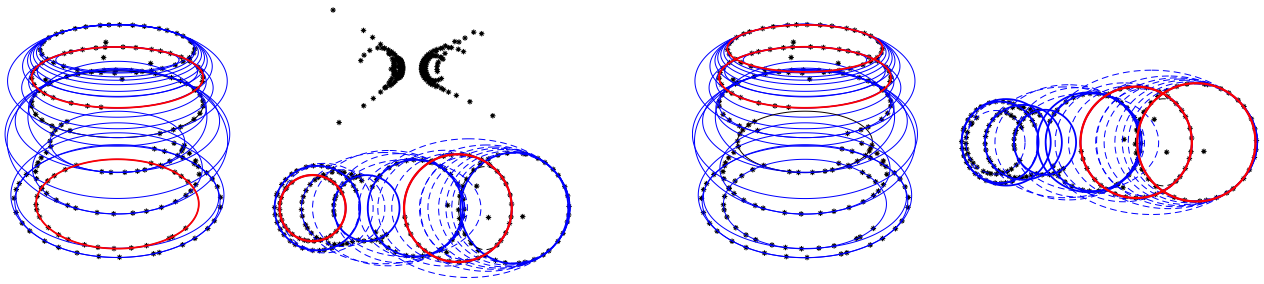
The solution of the robust regression problem is computed as follows. For each pair of ellipses $(\mathsf{A}_1, \mathsf{A}_2) \in E^2$, we count the number of ellipses of $E \setminus \{\mathsf{A}_1, \mathsf{A}_2\}$ passing through the intersections of $(\mathsf{A}_1, \mathsf{A}_2)$. The intersections contained by the largest number of ellipses of E are assumed to be the ICP of the supporting plane of the table.

The proposed solution for the identification of the ICP raises however the following two sub-problems.

Sub-problem 1 – The first sub-problem is how to count the number of ellipses of E passing through the intersections of a pair of ellipses. Two ellipses images of parallel 3D-circles intersect in a pair of real points and a pair of complex points or in two pairs of complex points, one of which is the ICP. In the presence of noise, the ellipses images of the vertices trajectories do not contain the ICP of the table supporting plane. More generally, these ellipses do not intersect in the same pair of points. For this reason, none of the ellipses pass through the ICP in practice. The complex projective space $\mathbb{CP}^2$ is not endowed with a metric that would allow us to use a threshold on the distance from a point to an ellipse, to determine whether or not an ellipse contains a given pair of complex conjugate points. We propose to circumvent this problem by considering that an ellipse A *intersects* the ICP if the ellipse $\mathsf{H}^\top \mathsf{A}\, \mathsf{H}$, rectifying A by some homography

$$\mathsf{H} = [\,\mathbf{h}_1 \;\; \mathbf{h}_2 \;\; *], \tag{1}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are the real and imaginary parts of one of the IPC, is a circle. Concretely, the rectification is considered to be a circle if the ratio of the semi-axes of $\mathsf{H}^\top \mathsf{A}\, \mathsf{H}$ is close to 1. For a pair of disjoint ellipses, two homographies composed of the real and imaginary parts of the two pairs of points are applied to the set of ellipses. The homography with the highest number of good rectifications is retained. In the case of a pair of ellipses with real intersections, only the complex conjugate intersections are tested (see Figure 3).



(a) (b)

Figure 3. Example of Euclidean rectifications of a set of points V (black asterisks) from a pair of ellipses, highlighted in red, out of all the K_{best} ellipses (in blue) from the exhaustive search. Ellipses rectified in circles are shown in solid line, in dashed line otherwise. Red circles are the rectifications of the pair of ellipses used to compute the rectifications. (a) The two red ellipses (left) do not intersect, hence two rectifications are therefore computed from their intersections, namely from the two pairs of conjugate complex points. Top: no circle is found. Bottom: four ellipses, highlighted in blue, are rectified into circles. (b) When the red ellipses intersect into two real points, only one rectification is computed. Five ellipses are then rectified into circles. The ellipses rectified into circles are images of polyhedron trajectories.

Sub-problem 2 – The second sub-problem is an excessive cost for computing the intersections of $|E|(|E|-1)/2$ pairs of ellipses of E. To address it, we argued in Subsection 4.2 that the ellipses of the family of imaged vertices trajectories are among the ellipses presenting the best scores. The number of intersection computations is thus reduced to $K_{\text{best}}(K_{\text{best}}-1)/2$, by only considering the K_{best} best ellipses. The value of K_{best} is important but not critical. Its empirically fixed value must be large enough to guarantee that the K_{best} best ellipses include the ellipses images of the trajectories of at least two distinct vertices. We choose $K_{\text{best}} = 5|V|$, which reduces the computation of the intersections to the solution of around $25|V|^2/2$ systems of two quadratic equations in two variables. A result of the ICP computation is shown in Figure 2b, where the pair of ellipses maximising the number of ellipses rectified into circles is shown in red in the rectified representation.

Partitioning – Instead of directly classifying the ellipses intersecting the ICP as belonging to the family of the imaged vertices trajectories and collecting the associated correspondences, Euclidean rectification to the points based on the knowledge of the ICP is first applied and the rectified points are partitioned into subsets of points located on circles. This strategy shows the best results. This can most likely be explained by a distribution of points in the rectified representation that is very well suited to partitioning in cases of ambiguous configurations, such as points distributed along portions of very close ellipses, as seen in the example at the top of Figure 2d, or very low angle shots.

Concretely, the points are rectified by the homography H^{-1} in its form (1). The circles formed by every point triplets of V in the rectified representation are computed. The circles are then sorted in descending order of the score described in Section 4.1. For each circle, processed by descending order of score, the subset of points in its vicinity is removed from V, until the number of points remaining is less than 5. The correspondence solution to our problem is provided by the obtained non-empty subsets of points.

Singularity B – The polyhedron might be positioned such that the images of the trajectories of two vertices lie on the same ellipse. The proposed method cannot separate the two subsets of correspondences in that case. In practice, the user can avoid such a configuration by off-centering the polyhedron with respect to the rotation axis of the table.

The 3D-reconstruction of the polyhedron and camera poses from the ICP $\mathbf{h}_1 \pm i\mathbf{h}_2$ and the imaged vertices correspondences delivered by our matching algorithm can then be performed, following the work of Jiang et al.[7] However, the provided set of correspondences may still comprise some outliers, namely silhouette vertices located on ellipses that are found and which are not imaged polyhedron vertices. The method of Jiang et al. does not explicitly address this problem, hence we modified it to make it robust. The details of the complete method can be found in the supplementary material.[19]

5. EXPERIMENTAL RESULTS

We conducted a large number of experiments with synthetic and real images to quantify the performance of the proposed method.

5.1 Synthetic Data

Synthetic polyhedra are generated as follows. Their vertices are obtained by computing the convex envelope of a set of 3D-points of arbitrary cardinality much greater than eight. The coordinates of the 3D-points are sampled randomly within $[-1, 1]^3$. Points are removed from the convex envelope until sets of 8, 7, 6, 5, and 4 points are obtained, which provides us with five sets of convex polyhedron vertices. The remaining points are scaled so that the three coordinates of all the points $\mathbf{X}_n$ belong to $[-1, 1]$. Forty polyhedra per number of vertices are generated which represents a total of $40 \times 5 = 200$ polyhedra.

The camera resolution is 1000×1000. The focal length is 2000 pixels and the principal point is set at the image center. The camera is located at a randomly selected distance from the centre of rotation of the turntable in the range $[6, 7]$. The camera points towards the table rotation axis. For each polyhedron geometry, five camera tilts are generated, randomly taken nearby a regular angle distribution between -15 and -50 degrees. A series of $J = 35$ angles θ^j of the turntable are randomly sampled according to a Gaussian distribution with 0 mean and a standard deviation of 3 degrees applied on angles regularly distributed between 0 and 360 degrees. Vertices reprojections are generated and, for each view, only those located on the convex hull of all the reprojections are considered as visible. White Gaussian noise of standard deviation varying from 0 to 2 pixels is added to both coordinates of the obtained reprojections. Outliers are introduced in the reprojections bounding box.

The results are expressed using the rate of *successful* matching and the root mean squared error (RMSE) between the estimated vertex positions and the ground truth ones. Given a polyhedron, matching is considered as *successful* if *(i)* the number of partitions found in the matching is equal to the number of ground truth partitions, namely the number of vertices, if *(ii)* there is no misclassified point, and *(iii)* at least three points per partition are found. Matching accuracy, namely the rate of correctly classified matches is also evaluated.

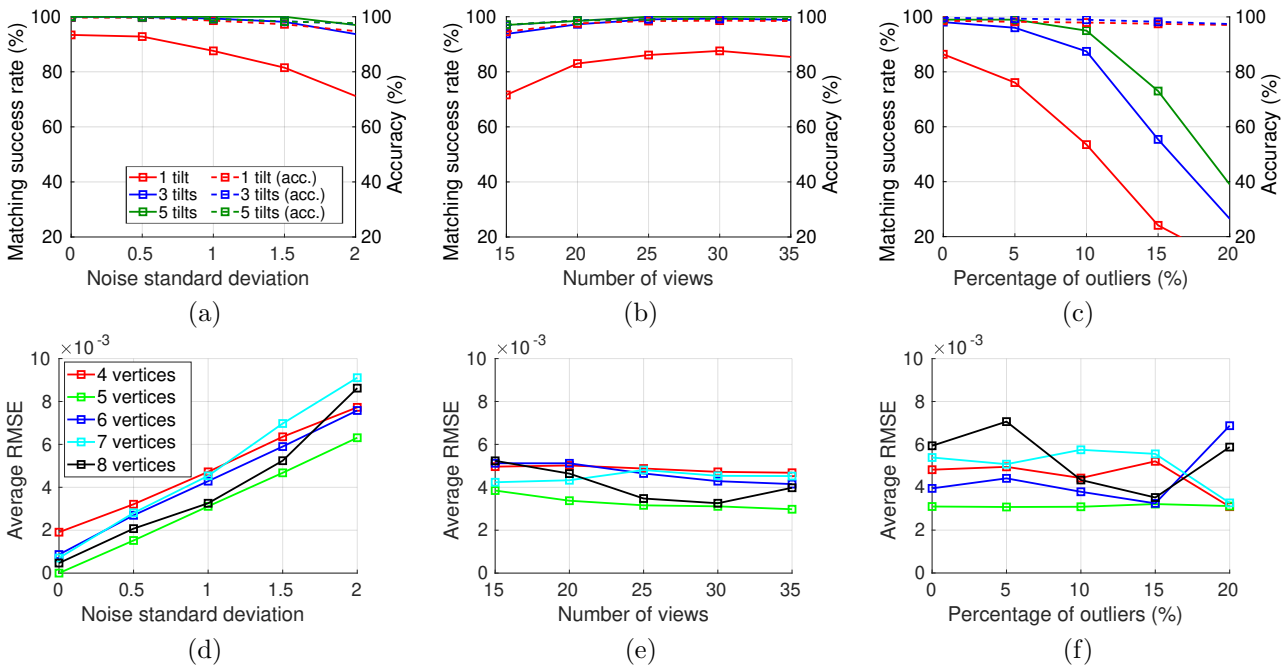


Figure 4. (a,b,c) Success rates (solid lines) and accuracy (dashed lines) of the proposed matching. (d,e,f) RMSE evaluated on the 3D-reconstruction performed when the matching is successful. RMSE reported in (f) has a low correlation with the percentage of outliers, as long as the robustness of the method is not compromised.

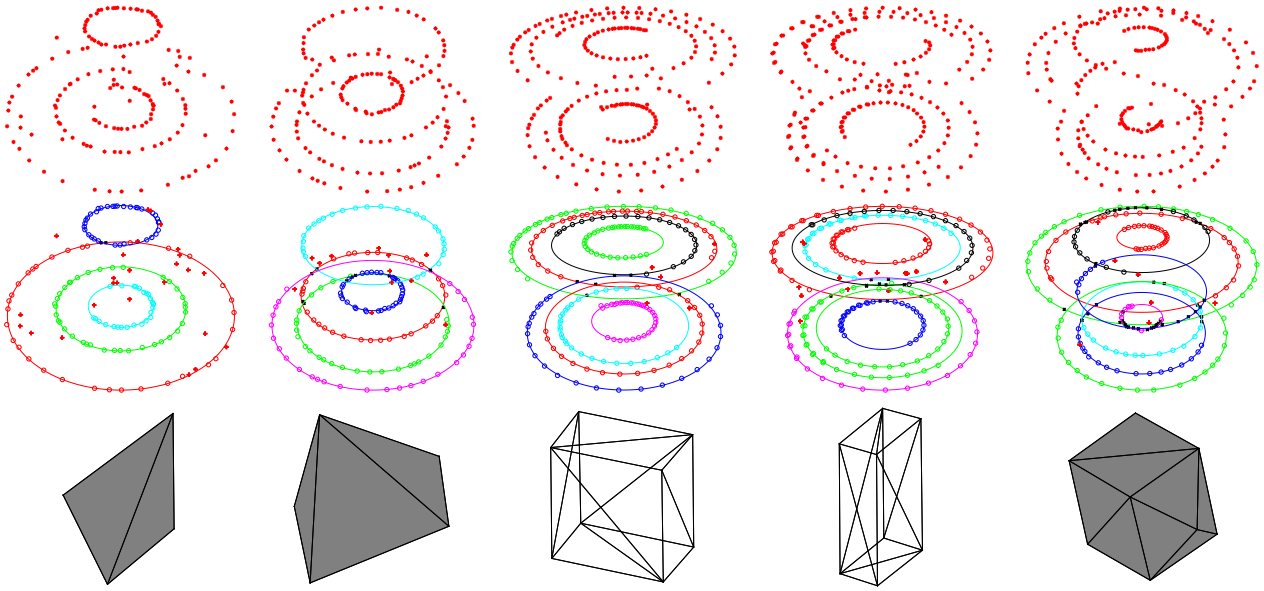
The rates of successful matching and matching accuracy are reported in Figure 5.1a-c, and RMSE evaluated over the successful matching cases in Figure 5.1d-f. They are functions of the level of noise and the number of views, evaluated with 3% of outliers, and function of the amount of outliers.

For each polyhedron, the method is run over all the camera tilts and the best solution is automatically selected. It is the one showing the lowest reprojection error. Acquiring a collection of views of the polyhedron with several camera tilts is straightforward in practice, for example using a standard camera tripod.

The matching shows overall a success rate greater than about 80% apart from the case with lowest number of views (15) and the highest level of noise (2 pixels) and reaches 100% for all polyhedra, from 25 to 35 views and from 0 to 1.5 pixels of noise with 5 camera tilts. Additional results evaluating the proposed ICP estimation method are provided in the supplementary material.[19]

5.2 Real Data

The entire pipeline was evaluated on five real datasets. The first dataset corresponds to the views of a tetrahedron which is the polyhedron with the minimum number of vertices, namely $N = 4$. The other datasets correspond to a square-based pyramid ($N = 5$), two parallelepipeds ($N = 8$), and a 9-vertex polyhedron. The two parallelepipeds are two transparent resin blocks, the first with an encased beetle, the second with an encased grasshopper. The other polyhedra are opaque. The view acquisitions are performed with a single camera tilt. The results are shown in Figure 5. The results of refractive MVS[22] from camera poses and polyhedron computed with markers[12] glued on the polyhedron and with our method are shown in Figure 6. It is therefore possible to use the marker-based results as a control reconstruction to assess the quality of the proposed reconstruction pipeline. MVS fails to reconstruct the resin block. The proposed method shows nearly identical refractive MVS results to the marker-based ones. Results of MVS without refraction[2] for the beetle, results for the grasshopper, and additional results showing that the method performs well under uncontrolled challenging lighting conditions are reported in the supplementary material.[19]



(a) (b) (c) (d) (e)

Figure 5. Top row: Set of points V input of the processing chain for the datasets ***tetrahedron***, ***square pyramid***, ***parallelepiped A***, ***parallelepiped B*** and ***9-vertex polyhedron***. Middle row: Results of the proposed matching. A colour is associated with each vertex. Red crosses indicate unclassified points and black ones indicate ambiguous correspondence class. Both are excluded from the input data for 3D-reconstruction. Bottom row: Solutions of the 3D-reconstructions. The matching and reconstruction succeeds on the five datasets.

6. CONCLUSION

We presented a pipeline for 3D-reconstruction of a convex polyhedron. The method is well suitable for poorly textured and non-Lambertian materials without controlled lighting. Robustness and accuracy of the method is evaluated on synthetic and real data. The method shows to perform well even for transparent medium.

We observed that the matching success drops for a number of vertices greater than 10. We plan to improve our method by adding a final guided matching step so as to retrieve correspondences that are missed using the estimated camera poses.

(a) (b) (c) (d)

Figure 6. (a-b) Excerpt of the input images of a resin block with an encased beetle. (a) Markers are placed on the block to compute a control reconstruction using Meshroom[2] with AprilTag.[12] (b) Block without markers used to evaluate our method. (c) Results of the 3D-reconstructions using refractive MVS[22] from camera poses and interface geometry computed with (c) markers, and (d) our method. The proposed method shows a reconstruction nearly identical to the one computed with the interface geometry obtained using markers. Standard MVS (Meshroom) fails to reconstruct the beetle with camera poses computed with markers (see supplementary material[19]).

REFERENCES

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

[2] Griwodz, C., Gasparini, S., Calvet, L., Gurdjos, P., Castan, F., Maujean, B., de Lillo, G., and Lanthony, Y., "Alicevision Meshroom: An open-source 3D reconstruction pipeline," in [*Proceedings of MMSys*], (2021).

[3] Schönberger, J. L. and Frahm, J.-M., "Structure-from-motion revisited," in [*Proceedings of CVPR*], (2016).

[4] Müller, T., Evans, A., Schied, C., and Keller, A., "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics* **41**(4) (2022).

[5] Fitzgibbon, A. W., Cross, G., and Zisserman, A., "Automatic 3D Model Construction for Turn-Table Sequences," in [*European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*], (1998).

[6] Jiang, G., Tsui, H.-T., Quan, L., and Zisserman, A., "Single Axis Geometry by Fitting Conics," in [*Proceedings of ECCV*], (2002).

[7] Jiang, G., Tsui, H.-T., Quan, L., and Zisserman, A., "Geometry of Single Axis Motions Using Conic Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10), 1343–1348 (2003).

[8] Zhang, H. and Wong, K. K., "Self-calibration of turntable sequences from silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(1), 5–14 (2009).

[9] Cheung, G. K. M., Baker, S., and Kanade, T., "Shape-from-silhouette across time part I: theory and algorithms," *International Journal of Computer Vision* **62**(3) (2004).

[10] Cheung, G. K. M., Baker, S., and Kanade, T., "Shape-from-silhouette across time part II: applications to human modeling and markerless motion tracking," *International Journal of Computer Vision* **63**(3) (2005).

[11] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J., "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition* **47**(6), 2280–2292 (2014).

[12] Krogius, M., Haggenmiller, A., and Olson, E., "Flexible layouts for fiducial tags," in [*Proceedings of IROS*], (2019).

[13] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R., "Nerf: Representing scenes as neural radiance fields for view synthesis," in [*Proceedings of ECCV*], (2020).

[14] Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M., "Pixelwise view selection for unstructured multi-view stereo," in [*Proceedings of ECCV*], (2016).

[15] DeTone, D., Malisiewicz, T., and Rabinovich, A., "Superpoint: Self-supervised interest point detection and description," in [*Proceedings of CVPR*], (2018).

[16] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[17] Ono, Y., Trulls, E., Fua, P., and Yi, K. M., "Lf-net: Learning local features from images," in [*Proceedings of NeurIPS*], Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., eds. (2018).

[18] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R., "ORB: An efficient alternative to SIFT or SURF," in [*Proceedings of ICCV*], (2011).

[19] Brument, B., Calvet, L., Bruneau, R., Mélou, J., Gasparini, S., Quéau, Y., Lauze, F., and Durou, J.-D., "A Shape-from-silhouette Method for 3D-reconstruction of a Convex Polyhedron – Supplementary material."

[20] Kanatani, K. and Ohta, N., "Automatic Detection Of Circular Objects By Ellipse Growing," *International Journal of Image and Graphics* **4**(1), 35–50 (2004).

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

[22] Cassidy, M., Mélou, J., Quéau, Y., Lauze, F., and Durou, J.-D., "Refractive multi-view stereo," in [*Proceedings of 3DV*], (2020).