IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 May 2024, accepted 27 May 2024, date of publication 30 May 2024, date of current version 6 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3407341

RESEARCH ARTICLE

Alignment and Improvement of Shape-From-Silhouette Reconstructed 3D Objects

ALBERTO J. PEREZ[1], JAVIER PEREZ-SOLER[2], JUAN-CARLOS PEREZ-CORTES[2], AND JOSE-LUIS GUARDIOLA[2]

[1]Departamento de Informática de Sistemas y Computadores (DISCA), Universitat Politècnica de València, 46022 Valencia, Spain
[2]Instituto Tecnológico de Informática (ITI), Universitat Politècnica de València, 46022 Valencia, Spain

Corresponding author: Alberto J. Perez (aperez@disca.upv.es)

This work was supported in part by European Union Horizon Europe Programme ''Artificial Intelligence Driven Industrial Equipment Product Life Cycle Boosting Agility, Sustainability and Resilience'' (AIDEAS) under Grant 101057294; and in part by the Generalitat Valenciana through Instituto Valenciano de Competitividad Empresarial [Valencian Institute of Business Competitiveness (IVACE)] Distributed Nominatively to Valencian Technological Innovation Centres under Project IMAMCA/2023/11.

ABSTRACT 3D object alignment is essential in multiple fields. For instance, to allow precise measurements in metrology, to perform surface/volumetric checks or quality control in industrial inspection, to align partial captures of a 3D object during object scanning, to simplify object recognition or classification in pattern recognition, accuracy and speed, being opposed, are desirable features of those algorithms. Nevertheless, they can be more or less critical depending on the application area. In the present work, we propose a methodology to improve the alignment of 3D objects reconstructed using shape-from-silhouette techniques. This reconstruction technique produces objects with small synthetic bulges, making them more difficult to align accurately. On the one hand, prealignment and branch-and-bound techniques are used to improve the convergence and speed of the alignment algorithms. On the other hand, a method to obtain a precise alignment even in the presence of bulges is presented. Finally, a refinement of the shape-from-silhouettes technique is shown. This technique uses multiple captures to refine object reconstruction and reduce or eliminate, among other improvements, synthetic bulges.

INDEX TERMS 3D alignment, 3D reconstruction, shape-from-silhouette, branch-and-bound, industrial inspection.

I. INTRODUCTION

3D reconstruction from images is a powerful group of techniques for capturing and representing the three-dimensional structure of objects and scenes using a set of two-dimensional images. Those techniques are widely used in fields such as computer vision, robotics, augmented reality, human pose estimation, and medical imaging [1], [2], [3], [4]. Binocular disparity, motion, silhouettes, linear perspective, atmosphere scattering, shading, textures, occlusions, and bilateral symmetries are among other features used to induce 3D structures from 2D images [2], [5].

The shape-from-silhouettes (SFS) approach [6], [7], [8] obtains a 3D model from the silhouettes of an object obtained from images taken from different positions. This method does not require detailed texture information, making it applicable in scenarios where texture information is lacking or unreliable. However, it relies heavily on an accurate silhouette extraction and camera calibration for successful reconstruction [7], [9], [10]. This technique is currently applied in several fields, such as industrial inspection [11], [12], human pose recognition [13], [14], medical imaging [4], [15].

In SFS, the object's 3D shape is constructed through the intersection of silhouette cones derived from multiple images. Each silhouette cone originated from the union of the

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

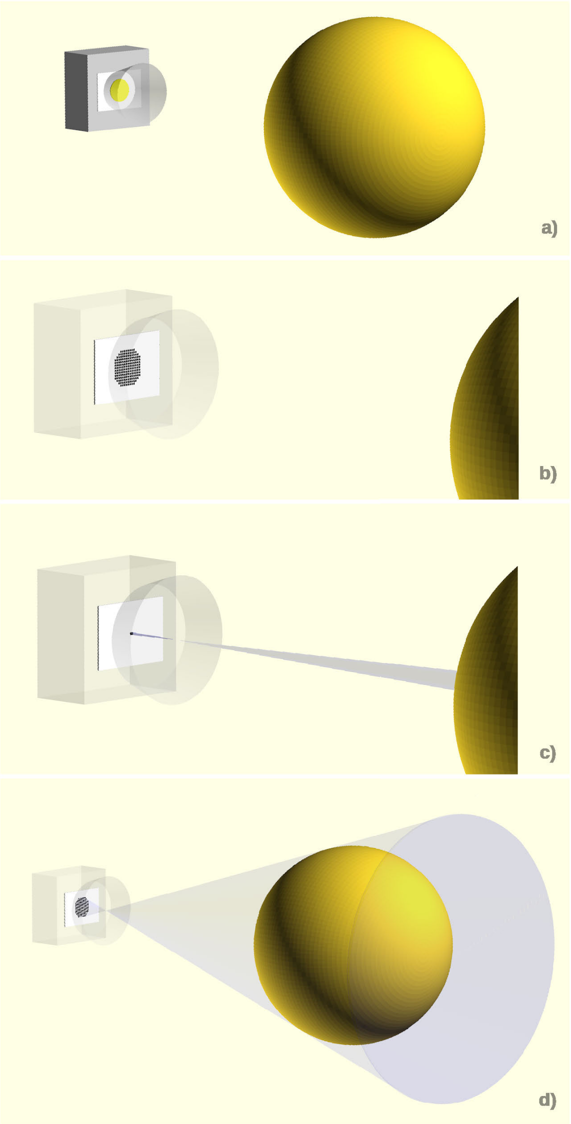


FIGURE 1. Generation of the silhouette cone of an object for each camera involves the following steps: First, an image of the object is captured (a). Next, the object is segmented in the image to produce a binary silhouette. Finally, a silhouette cone (d) is created by combining the projection cones of all the "object" pixels (c) within the silhouette, using the camera parameters.

projection of pixels identified as ''object'' in the segmented image (Figure 1). The intersection of these silhouette cones represents the visual hull [16], defining the largest shape consistent with the object's silhouettes observed from any viewpoint within a specified area. This computation relies on intrinsic and extrinsic camera parameters [6]. Hence, as previously noted, ensuring accurate segmentation and camera calibration is essential for maintaining reconstruction precision.

To obtain the visual hull, first an octree structure is generated by a carving process of an initial 3D cube using the space outside the silhouette cone of each image [17]. Next a polygonization of the octree structure is performed by using a marching cubes algorithm [18].

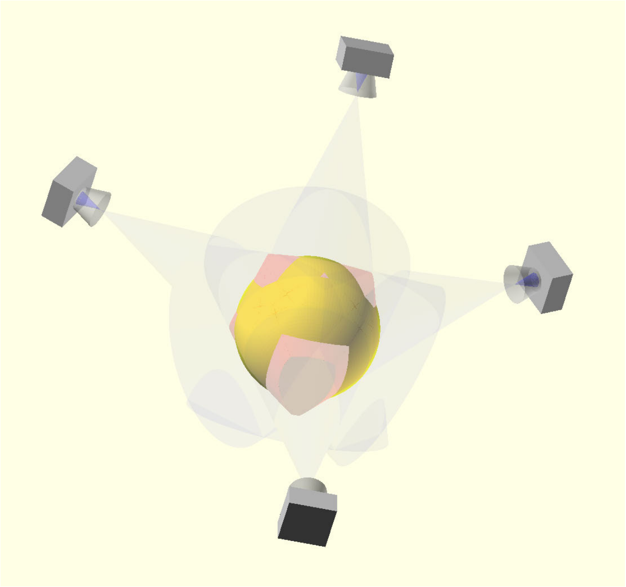


FIGURE 2. Visual hull obtained from silhouette cone intersections. Synthetic bulges appear on the object reconstruction (in red) depending on the number of cameras and their positions.

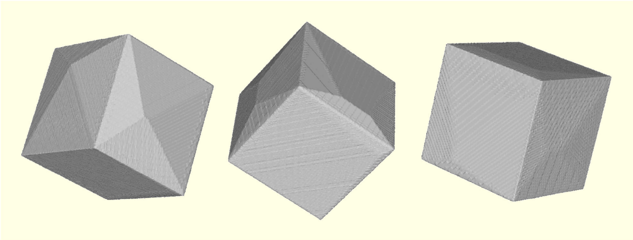


FIGURE 3. Cube reconstruction from 16, 24, and 48 cameras (left to right).

This technique allows to reconstruct the 3D shape of an object only from images taken from calibrated cameras positioned around the object. The more cameras are used, the better the reconstruction accuracy (Figure 3).

Although simple conceptually, this reconstruction method presents two main drawbacks. On the one hand, the concave surface regions can never be distinguished using silhouette information alone, thus making this unsuitable for those objects a priori. On the other hand, synthetic bulges can appear depending on the number of cameras and their positions (see Figure 2). Those bulges can complicate the alignment operations critical in many applications: metrology, industrial inspection, quality control, 3d object recognition, and classification.

This work presents several methodologies to address synthetic bulges and 3D alignment. A technique is proposed to improve reconstruction accuracy in shape-from-silhouette methods by utilizing multiple capture sets.

The paper is organized as follows: section II presents other related works, section III introduces prealignment techniques to obtain good initial guesses on ICP, section IV presents an accurate method to align reference models and reconstructed objects. In section V, a methodology is shown to refine reconstruction using different sets of captures of the same object. In section VI, some results are presented, in section VII the results are summarized, and finally, in section VIII the conclusions of our work are exposed.

II. RELATED WORKS

A. ALIGNMENT

Among the alignment algorithms, the Point-to-Plane Iterative Closest Point (ICP) [19], [20] offers in general a precise, robust, and efficient solution to match rigid surfaces or point clouds if a good initial guess is provided. In presence of noise, Sparse ICP [21] or EM-ICP [22] can be employed. Nevertheless, those methods require more computational time and some extra parameters have to be estimated to reject outliers correctly or to estimate the existing noise [23].

Featured-based alignment [24], [25] tries to identify surface or geometrical features to establish a constellation of features that allow an alignment transformation to be found. Those methods have substantial limitations if no texture or distinguishable features exist. The computational cost of the feature search can be high if the features are complex.

Other strategies based on Principal Component Analysis (PCA) [26], Deep Learning [27] or object symmetries [28], [29] do not need a coarse alignment or a good alignment initial guess, as ICP methods, but they do not provide, in general, an accurate alignment. Those strategies can be useful to obtain an initial guess for the ICP algorithms or for others tasks where an exact matching is not necessary, for example for object classification or recognition.

B. SHAPE-FROM-SILHOUETTES

The SFS method belongs to the multi-view reconstruction methods. Those methods try to reconstruct the 3d structure of an object based on 2D images. Among those methods we can find:

- Structure-from-Motion (SFM) [30], [31], [32]: object features must be identified in different captures where an object is in motion or the camera moves. A matching process uses those features and the camera model to reconstruct the scene.
- Multi-view Stereo (MVS) [33]: using images from two different calibrated cameras, object features are identified and triangulated to create a point-cloud representation of the object or surfaces using, for example, photoconsistency [34].
- Deep Learning methods (DL): deep convolutional neural networks (CNN) for stereo reconstruction (DeepMVS [35]) or visual hull learning (SiINET [36]). Objects to reconstruct have to be presented to the CNN first.

As commented before, SFS methods do not require texturized objects to infer the 3D structure of an object, as SFM or MVS methods or even DeepMVS, because they are based on silhouettes. Besides, computing silhouettes is, in general, a more straightforward and less time consuming process than searching and identifying features on images.

Nevertheless, inconsistent silhouettes pose a potential challenge, mainly when dealing with poor calibration or inaccurate or noisy silhouettes [10]. The reconstruction quality hinges on several factors, including calibration precision, silhouette accuracy, and the quantity of cameras employed. It's worth noting that the number of cameras utilized directly impacts the occurrence of bulges, as previously discussed.

Our work was mainly developed in the context of industrial inspection using a well-calibrated device [11]. The described device reconstructs free-falling objects using the images taken by a constellation of 16 cameras. Light conditions and background are controlled. Thus, object segmentation can be done accurately.

III. PREALIGNMENT

The convergence speed of aligning 3D objects using ICP depends on the initial object orientations. The more the orientations differ, the more computation power and time are required. It is possible even that the ICP does not converge in some situations, typically if orientations are too different [37] being unable to align the objects. To avoid this problem and to reduce computational power, rough alignment techniques are commonly employed to obtain initial guesses (Section II-A).

Some of those techniques offer one or more hypotheses that the ICP should explore to ensure convergence. That implies more computational cost, which will be addressed later in this section.

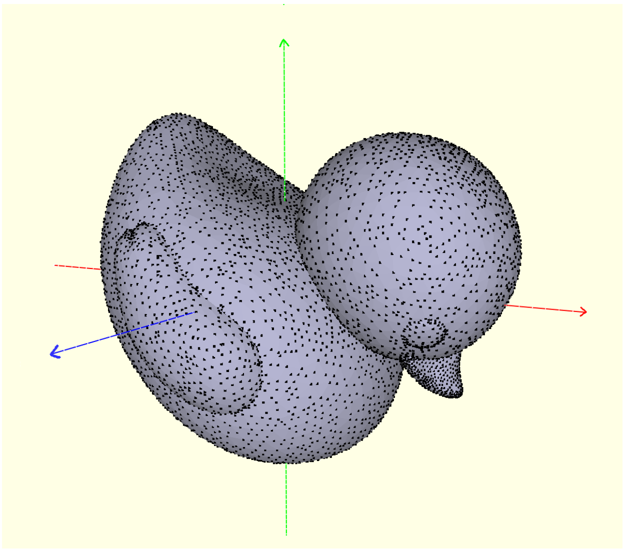


FIGURE 4. Principal axis computed by PCA over the point cloud of a 3D object: $\bar{v}_1$ (red), $\bar{v}_2$ (green) and $\bar{v}_3$ (blue).

In the present work, prealignment based on PCA is employed because it is straightforward and fast to compute from a point-cloud representation of a 3d object [26]. The PCA analysis gives three orthogonal axes (principal axis) representing the directions where the object points present maximal variance. As shown in Figure 4, aligning an object along its principal axis offers a fast method to normalize object orientation and, thus, simplify object alignment. This approximation can not achieve a precise alignment if objects

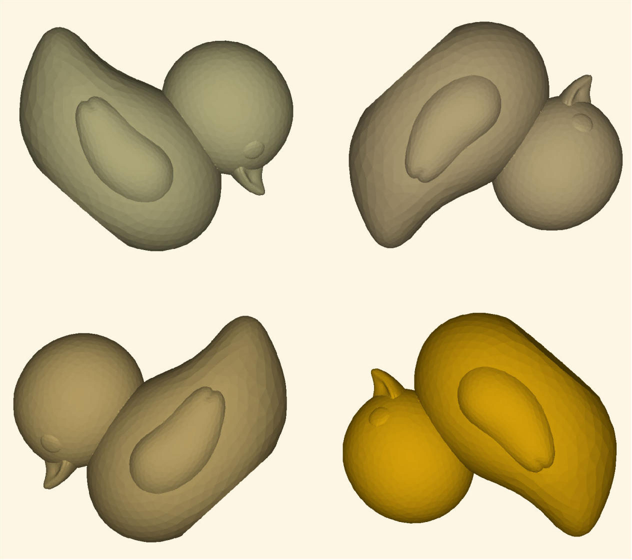


FIGURE 5. After computing PCA, four prealignment hypotheses have to be considered, taking into account both orientations of $\vec{v}_1$ and $\vec{v}_2$.

present defects, acquisition errors, noise, or, as in our case, reconstruction bulges. For this purpose, the ICP algorithm is employed.

It is important to note that the principal axes represent the maximal variance directions; thus, if objects are not symmetric, both orientations must be considered for the two first principal axes, this leads to 4 possible prealignment hypotheses.

Being $\vec{v}_1$,$\vec{v}_2$ and $\vec{v}_3$ the eigenvectors (principal axis) computed for the point set of an object with eigenvalues $e_1 > e_2 > e_3$, the object can be aligned using the transformation matrix,

$$h_1 = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vec{v}_3 \end{bmatrix}$$

considering each possible orientation of $\vec{v}_1$ and $\vec{v}_2$, the following transformation must be considered equally (see Figure 5),

$$h_2 = \begin{bmatrix} \vec{v}_1 \\ -\vec{v}_2 \\ -\vec{v}_3 \end{bmatrix}, h_3 = \begin{bmatrix} -\vec{v}_1 \\ \vec{v}_2 \\ -\vec{v}_3 \end{bmatrix}, h_4 = \begin{bmatrix} -\vec{v}_1 \\ -\vec{v}_2 \\ \vec{v}_3 \end{bmatrix}$$

Principal axes are ordered by its eigenvalues that represent the variance explained in each axis, but if two or more eigenvalues are similar, several hypotheses must be considered because the order is not defined, and eigenvectors can be selected in several combinations (see Table 1). For example in the Figure 6 eigenvalues are $e_1 \approx e_2 \approx e_3$ because principal axes have similar variance.

The ICP (Iterative Closest Point) algorithm iterates to obtain the rotation transform that aligns a pair of point sets, which, in our case, is obtained from a couple of objects. Each iteration minimizes a simplified and linearised expression of the quadratic error [19] using least squares until convergence.

TABLE 1. PCA hypothesis to consider in non-symmetric objects depending on the eigenvalues.

eigenvalues	hypothesis number
$e_1 > e_2 > e_3$	4
$e_1 > e_2 = e_3$	8
$e_1 = e_2 > e_3$	8
$e_1 = e_2 = e_3$	24

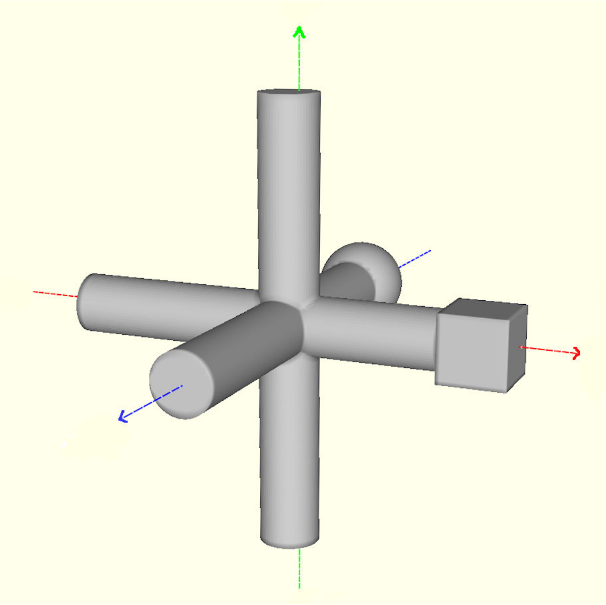


FIGURE 6. The eigenvalues of this object's principal axis (e_i) give a similar value. Thus, 24 hypotheses should be considered if the object is not symmetric.

The simplified expression is valid if objects are no *too much* misaligned. For this reason, a good initial guess or prealign hypothesis is necessary to obtain good convergence.

Every hypothesis has to be evaluated by the ICP if objects are not symmetrical; thus, the alignment cost can increase significantly. To minimize the cost of evaluating every hypothesis, the authors propose a branch and bound algorithm to prune the hypotheses that are not converging fast enough. During a few numbers of iterations, I_0 (one or two are enough in our experiments), all hypotheses are considered in parallel. Then, the best result is used to compute a bound ($err * B$), and hypotheses performing worst are pruned (See Algorithm 1).

Those techniques solve orientation. Translation is easily solved bringing the object's center of mass to the origin.

IV. ACCURATE ALIGNMENT WITH BULGES

The presence of bulges complicates ICP task because their location depends on the object's orientation during capture. A good designed capture system will try to minimize those artifacts but in some situations that can affect the alignment. For example, if a captured object has to be aligned with its CAD reference to check dimensions, surface defects, orientation, or whatever (see Figure 7) bulges in the captured image can reduce alignement precission.

Algorithm 1 B&B ICP

```
INPUT: $Obj, Ref, \Delta Err, B, I_0, I_T$
$Obj' = PCA(Obj)$
$\{Ref'_i : i = 1 \ldots N_H\} = HypothesysPCA(Ref)$
$err_0 = 0$
repeat
    $n = n + 1$
    for all $Ref'_i$ do
        $\{err_{i_n}, Ref'_i\} = ICP_{iteration}(Obj', Ref'_i)$
    end for
until $i < I_o$
repeat
    $err = \min_{i=1 \ldots N_H}(err_{i_n})$
    $n = n + 1$
    for all $Ref'_i$ do
        if $\|err_{i_n} - err_{i_{n-1}}\| > \Delta Err$ then
            if $err_{i_n} < B * err$ then
                $\{Ref'_i, err_{i_n}\} = ICP_{iteration}(Obj', Ref'_i)$
            else
                {Hypothesis Prunned}
            end if
        else
            {Hypothesis Converged}
        end if
    end for
until $i < I_T$
$s = arg \min_{i=1 \ldots N_H}(err_{i_n})$
OUTPUT: $Ref'_s$
```

To solve this problem, an iterative ICP has been proposed (see Algorithm 2). Exact matching is not possible because of the bulges, then after finding the affine transformation (M) that best align the reference object (Ref) with the reconstructed object (Obj) using ICP, a virtual set of images (I_{Ref}) of the reference object in the aligned position (Ref') is obtained. This can be done using the calibration parameters ($Calib$) of the camera setup used to reconstruct the object and using z-buffer techniques [38] to project the reference on the cameras. This set of images allows us to create a reconstructed version of the reference (Ref'_r) that will present bulges more or less in the same positions that the reconstructed object, and thus applying ICP again with the new reference a more precise alignment should be obtained. This is repeated until convergence.

V. REFINEMENT FROM MULTIPLE CAPTURES

The most important drawback of the shape-from-silhouette reconstruction method is the presence of bulges. Their presence reduces the accuracy of measurements and complicates alignment, but more interestingly, it avoids the possibility of obtaining precise models of captured objects.

As commented, those bulges can be minimized using more cameras (see Figure 3) but increasing the number of cameras is not always feasible due to cost or complexity reasons. Obtaining more images from a moving camera implies, on the one hand, more capture time and, on the other hand, and more importantly, it adds complexity because of the camera position and orientation have to be known very accurately for the reconstruction algorithm to work [11].

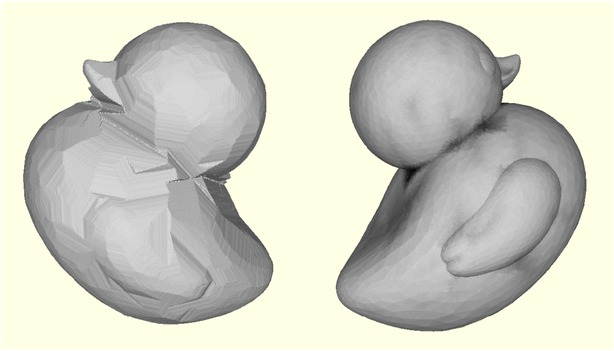


FIGURE 7. Capture bulges (left) can reduce alignment precision when aligning with CAD models (right).

Algorithm 2 Iterative ICP

```
INPUT: $Obj, Ref, Calib, \Delta Err$
$n = 0$
$\{M, err_n\} = ICP_{transf}(Obj, Ref)$
$Ref' = Transf(Ref, M)$
repeat
    $n = n + 1$
    $I_{Ref'} = \{i_1, i_2, \ldots, i_c\} = Z_{buffer}(Calib, Ref')$
    $Ref'_r = Reconstruct3D(I_{Ref'})$
    $\{M_r, err_n\} = ICP_ttransf(Obj, Ref'_r)$
    $Ref' = Transf(Ref, M_r)$
until $|err_n - err_{n-1}| < \Delta Err$
OUTPUT: $Ref'$
```

In this section, we propose to refine reconstruction using several sets of captures of the same object. We assume objects are captured in different position each time, either because they are presented to the capture system so or, as in [11], because they are captured on free fall through the system.

As explained in Section I, for each set of captures, an octree structure is generated by carving an initial 3D cube using the outside of the silhouette cone of each image (see Figure 9). With each image, the reconstruction of the captured object is refined. The exact camera positions, orientations, and intrinsic parameters must be known to compute the silhouette cone. Those are obtained in a calibration process for a real system [11].

A second set of images can not be used to keep carving the octree because the object is in a different position and the carving process will not erode in the right places. Nevertheless, we can align the object reconstructions obtained from each set, change the camera position in one of them to match the object's position and orientation, and redo the carving process with both sets of images and the new camera parameters (see Figure 8). The The result will resemble that of a system with double cameras. In the same way, several sets can be used to refine iteratively the reconstruction of an object, allowing us to eliminate bulges with no extra hardware.

The method works as follow (Algorithm 3): being $\{I_1, I_2, \ldots, I_k\}$, k sets of captures of the same object, k reconstructions are performed obtaining $\{Obj_1, Obj_2, \ldots, Obj_k\}$, a set of reconstructed objects. If a CAD version of the object exists, it can be used as a reference; if this is not the case, the first object, Obj_1, is chosen without loss of generality. Each object Obj_i is then aligned with the reference obtaining an affine transformation (M_i) user to modify the camera position for captures I_i (see Figure 8) to match the point of view of the reference. Initializing the set $C_i = \{[R_1, P_1], [R_2, P_2], \ldots, [R_c, P_c]\}$ with the original orientations (R_j) and positions (P_j) of each camera (calibration data), the values for the capture set i are recomputed as,

$$R'_j = [M_i R_j^T]^T$$
$$P'_j = M_i P_j$$

Using the pairs $\{[I_i, C_i] \mid i \in [1, k]\}$, a new octree is computed obtaining information from all the capture sets. The new octree is polygonized using the marching cubes algorithm [18] to obtain a 3D object as in the simple case.

Algorithm 3 SFS Refinement

```
INPUT: $\{I_1, I_2, \ldots, I_k\}$
INPUT: $Calib = \{[R_1, P_1], [R_2, P_2], \ldots, [R_c, P_c]\}$
for all $I_i$ do
    $Obj_i = Reconstruct3D([I_i, Calib]$
end for
if CAD$model$ then
    $Ref$ = CAD$model$
else
    $Ref = Obj_1$
end if
for all $Obj_i$ do
    $M_i = ICP_t transf(Obj_i, Ref)$
    for $j = 1 \ldots c$ do
        $R'_j = [M_i R_j^T]^T$
        $P'_j = [M_i P_j]$
    end for
    $C_i = \{[R'_1, P'_1], [R'_2, P'_2], \ldots, [R'_c, P'_c]\}$
end for
$Obj = Reconstruct3D([I_1, C_1], [I_2, C_2], ), \ldots [I_k, C_k])$
OUTPUT: $Obj$
```

When implemented to reduce memory usage, the process can be done iteratively using a capture set each time, reconstructing, aligning, modifying camera calibration, refining the octree, and then discarding all this information for the next iteration with a new capture set.

VI. EXPERIMENTS

To test the proposed algorithms, a set of synthetic objects (See Figure 10 and Table 2) is used. A model of a capture device (see Figure 11 [11]) is defined, and the synthetic capture sets were obtained using a z-buffer techniques [38]

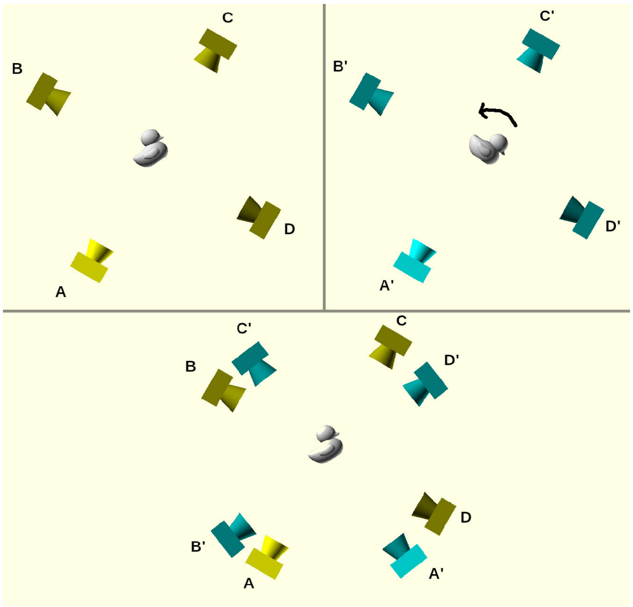


FIGURE 8. In a system with four cameras, two sets of captures of the same object in different orientations are taken (top). After aligning both reconstructions (black arrow), one of the camera sets is reoriented accordingly, and a new reconstruction can be performed using both capture sets (bottom).

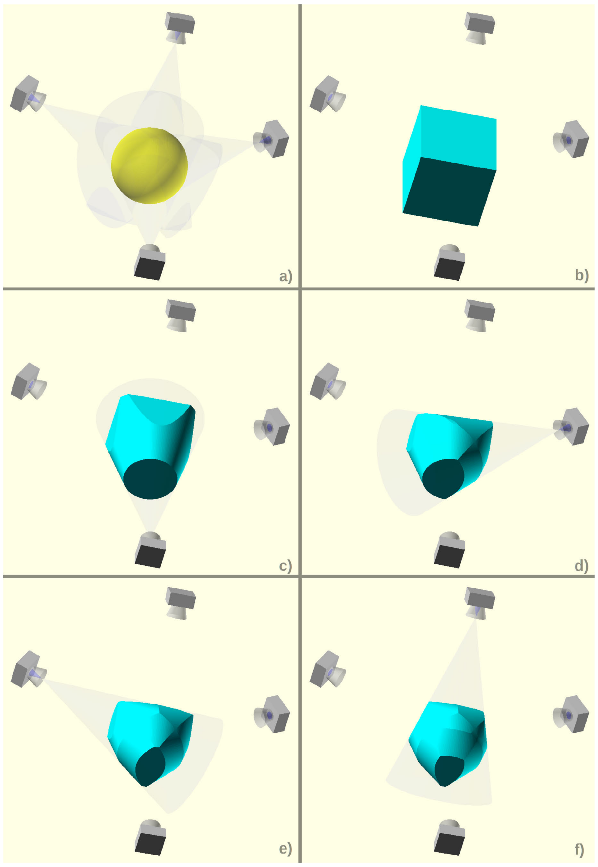


FIGURE 9. A carving process is performed with the silhouette cone of each camera over a 3d cube. A sphere is captured (a), the initial cube (b) and the carving process (c)-(f).

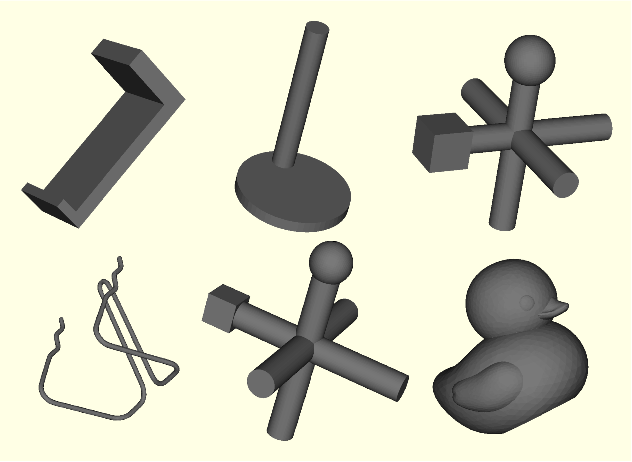


FIGURE 10. Test set of objects: ***hard A, hard C, ninja A, spring, ninja B and duck*** (left-right, up-bottom).

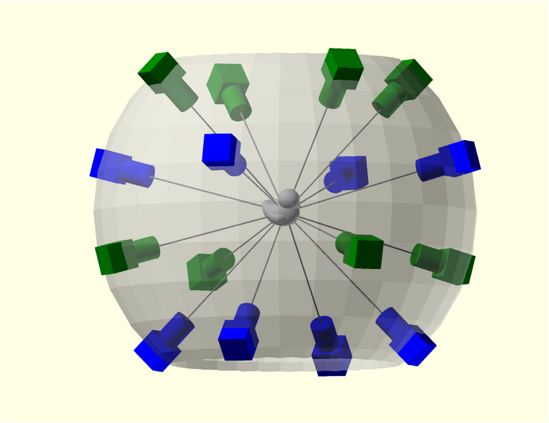


FIGURE 11. Model of the capture system. A constellation of 16 cameras is arranged in a sphere pointing to its center.

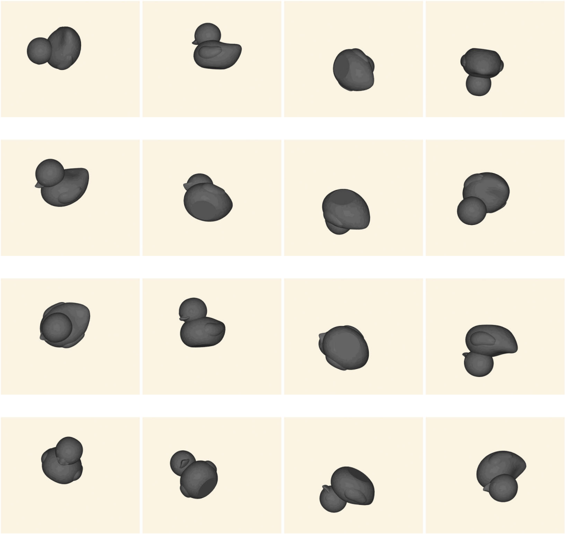
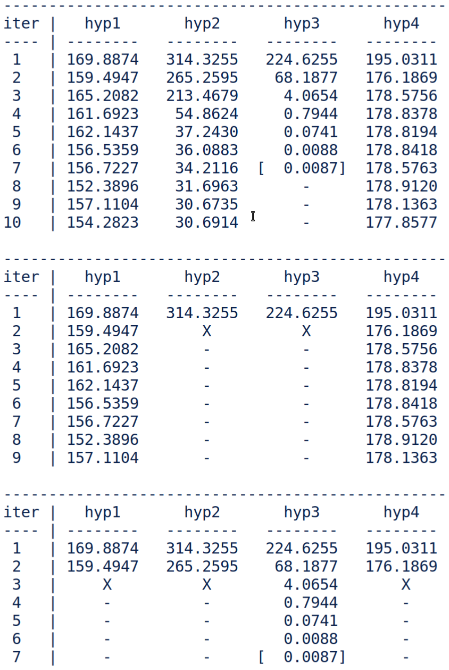


FIGURE 12. An example of synthetic capture set.

(see Figure 12). In the model, the cameras are arranged in a sphere of radius 560 cm, with 2D sensor of 2448 × 2048 pixels of size 3.45μm and with optics of focal length of 50 mm.

TABLE 2. Synthetic objects statistics.

Object	# triangles	# points	size (mm)
duck	77278	38615	75x60x64
ninja A	167142	83573	30x30x30
ninja B	220546	110275	30x30x30
spring	488642	244317	110x110x65
hard A	110592	55298	34x15x15
hard C	91696	45850	20x20x30



```
---------------------------------------------------
iter |   hyp1       hyp2       hyp3       hyp4
---- | --------   --------   --------   --------
 1   | 169.8874   314.3255   224.6255   195.0311
 2   | 159.4947   265.2595    68.1877   176.1869
 3   | 165.2082   213.4679     4.0654   178.5756
 4   | 161.6923    54.8624     0.7944   178.8378
 5   | 162.1437    37.2430     0.0741   178.8194
 6   | 156.5359    36.0883     0.0088   178.8418
 7   | 156.7227    34.2116   [ 0.0087]  178.5763
 8   | 152.3896    31.6963       -      178.9120
 9   | 157.1104    30.6735       -      178.1363
10   | 154.2823    30.6914       -      177.8577

---------------------------------------------------
iter |   hyp1       hyp2       hyp3       hyp4
---- | --------   --------   --------   --------
 1   | 169.8874   314.3255   224.6255   195.0311
 2   | 159.4947      X          X       176.1869
 3   | 165.2082      -          -       178.5756
 4   | 161.6923      -          -       178.8378
 5   | 162.1437      -          -       178.8194
 6   | 156.5359      -          -       178.8418
 7   | 156.7227      -          -       178.5763
 8   | 152.3896      -          -       178.9120
 9   | 157.1104      -          -       178.1363

---------------------------------------------------
iter |   hyp1       hyp2       hyp3       hyp4
---- | --------   --------   --------   --------
 1   | 169.8874   314.3255   224.6255   195.0311
 2   | 159.4947   265.2595    68.1877   176.1869
 3   |    X          X         4.0654      X
 4   |    -          -         0.7944      -
 5   |    -          -         0.0741      -
 6   |    -          -         0.0088      -
 7   |    -          -       [ 0.0087]     -
```

FIGURE 13. Evolution of the different versions of the ICP algorithm for the object *spring*: standard version (top), bound after the first iteration (center), bound after the second iteration (bottom). A value in brackets means minimal error attained. A X means hypothesis bounded.

For each object, 20 synthetic sets of captures are generated. Each object was presented in a random position and orientation near the center of the capture system. Position and orientation were saved as ground truth for our experiments. It is worth commenting that synthetic captures are used, among other reasons, because position and orientation could not be available for real captures.

Using those capture sets, 20 object reconstructions have been computed using the shape-from-silhouette method; bulges appear on them in different positions. Each reconstruction was labeled with its position and orientation.

A. PREALIGNEMENT AND BOUNDED ICP

Figure 13 shows the evolution of the ICP algorithm while considering simultaneously the 4 PCA hypothesis for the object *spring*. Hypothesis 3 obtains the correct alignment, while the others lead to incorrect alignment because the ICP algorithm reaches local minima. This example shows the importance of starting ICP with a good guess, as stated in [37]. The same example illustrates the importance of computing enough iterations before starting pruning in the bounded ICP. In Figure 13, it can be seen that pruning after

TABLE 3. Simple versus bounded ICP: alignment error and computational time.

Object	Non-Bounded				Bounded			
	Error (deg.)		Time (sec.)		Error (deg.)		Time (sec.)	
	mean	std	mean	std	mean	std	mean	std
duck	0.230	0.093	0.240	0.025	0.230	0.093	0.101	0.004
ninja A	0.141	0.024	0.250	0.041	0.141	0.024	0.141	0.011
ninja B	0.091	0.021	0.276	0.033	0.090	0.022	0.174	0.018
spring	0.030	0.003	0.490	0.033	0.030	0.003	0.255	0.012
hard A	0.469	0.235	0.137	0.028	0.469	0.235	0.072	0.006
hard C	1.264	0.667	0.473	0.039	1.264	0.667	0.112	0.009

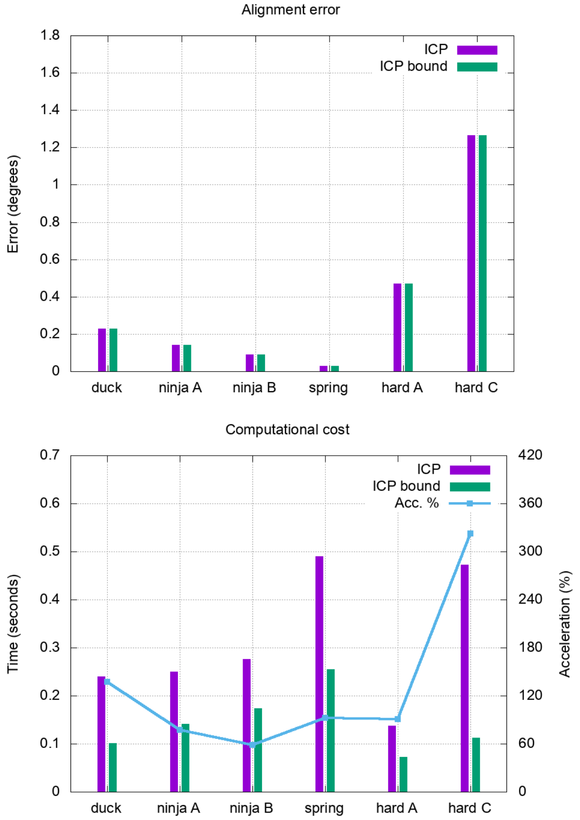


FIGURE 14. Alignment errors and computational time comparison: simple and bounded ICP.

the first iteration ($I_o = 1$) eliminates the correct hypothesis; in this case, at least two iterations are necessary to allow the ICP to converge adequately. In our experiments, an I_0 value of 2 was enough to assure convergence for all the objects.

To test the algorithms, each reconstructed object has been aligned against its original reference using both simple ICP and bounded ICP. Being M_o the labeled rotation, and M_i the rotation obtained with the ICP, the alignment error is defined as the difference transform angle, a_{err}, computed from the difference transform, M_{diff}, as follows [39],

$$\begin{aligned} M_{diff} &= M_o M_i^{-1} \iff M_{diff} * M_i = M_o \\ a_{err} &= (180.0/\pi) * acos(tr(M_{diff}) - 1)/2)) \end{aligned}$$

The results can be seen in Table 3 and Figure 14. The alignment error mean and standard deviation among the 20 reconstructions are presented for each object. Alignment errors are similar for the standard and the bounded ICP, while the bounded ICP has a smaller computational cost. Objects *hardA* and *hardC* present more significant errors and dispersion. In part because bulges complicate alignment and in part because of their geometry.

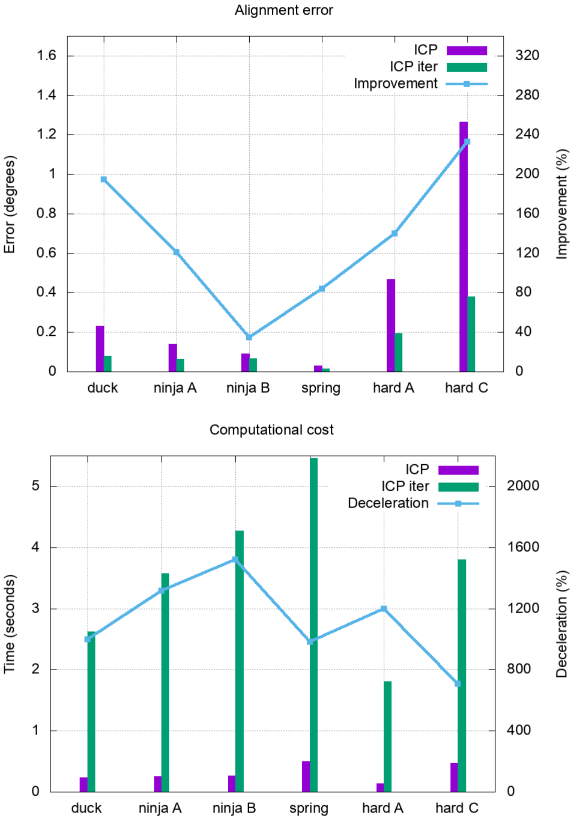


FIGURE 15. Alignment errors and computationa time on bounded ICP.

B. ACCURATE ALIGNMENT WITH BULGES

As in the previous section, iterative ICP has been applied to the reconstructed objects, and alignment error statistics have been calculated. Table 4 and Figure 15 present the results. The iterative algorithm significantly improves the alignment error mean for every object. Objects such as *duck* or *hardC* obtain the best results, with improvements over 200% error alignment. Dispersion is also greatly reduced for those objects and *hardA*. Removing the influence of bulges, both alignment errors and result dispersion are improved.

Nevertheless, the computational cost increases significantly. On the one hand because the alignment is done several times, a mean of 5 iterations are needed for the algorithm to converge. And on the other hand, because of the extra cost of computing the projection and reconstruction of the reference with bulges in each iteration.

C. MULTIPLES CAPTURES

In this section, we show how the multicapture reconstruction improves object reconstruction just in a couple of cases.

TABLE 4. Iterative alignement errors and computational time.

Object	Iterative			
	Error (deg.)		Time (sec.)	
	mean	std	mean	std
duck	0.078	0.033	2.622	0.077
ninja A	0.064	0.028	3.579	0.090
ninja B	0.068	0.028	4.272	0.068
spring	0.016	0.005	5.462	0.228
hard A	0.196	0.131	1.808	0.093
hard C	0.380	0.214	3.799	2.949

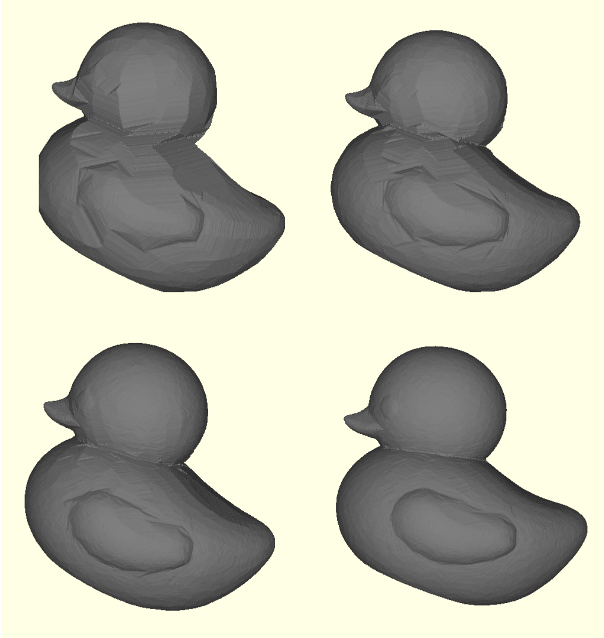


FIGURE 16. Object reconstruction improvement through multiple capture refinement. (left to right and top to bottom) One capture, two captures, four captures, and 16 captures).

In Figures 16 and 17 it is possible to appreciate qualitatively the improvement obtained using 1, 2, 4, and 16 captures of the same object for the case of the object *duck* and for a new object, *tube*. Especially interesting is the case of *tube* the hole through the object can be reconstructed accurately using several captures.

VII. DISCUSSION

The presented bounded ICP has a smaller computational cost than the standard ICP while offering the same alignment precision. This approach allows us to consider more prealignment hypotheses to obtain a better precision and convergence width for the ICP standard algorithm.

On the other hand, the iterative ICP algorithm significantly improves the alignment error for every tested object, minimizing or removing the influence of bulges on SFS reconstructed objects. This improvement is obtained at the expense of a higher computational cost, but if alignment accuracy is important, the extra cost is justified.

Finally, the proposed SFS refinement method allows to control the reconstruction accuracy if several sets of captures of the same object in different positions can be obtained. With each new set of captures, the reconstruction can be refined reducing or eliminating bulges. This method allows a more accurate SFS reconstruction with less hardware cost.

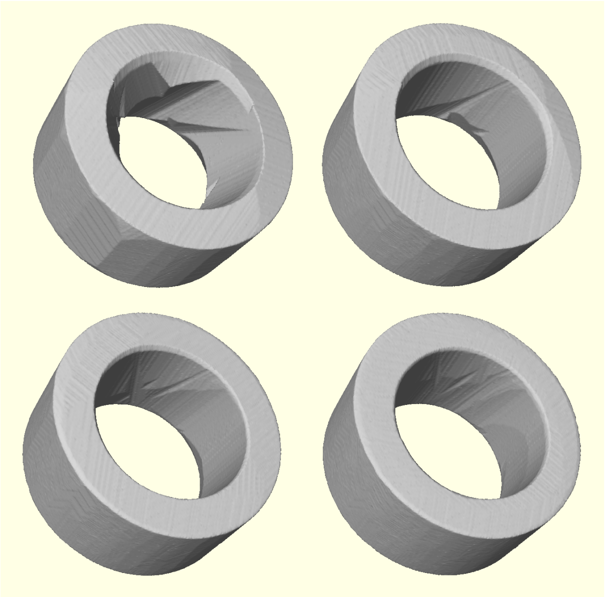


FIGURE 17. Another example of object reconstruction improvement through multiple captures (left to right and top to bottom: 1, 2, 4, and 16 captures).

VIII. CONCLUSION

In the present work, several techniques to work with objects reconstructed using the *shape-from-silhouettes* method have been presented.

First, a bounded version of the ICP alignment algorithm was presented to speed up alignment when several hypotheses have to be considered. Experiments in Section VI-A show that the algorithm can reduce the computational cost, but special attention should be paid considering in which iteration (I_0) the bound process starts to avoid pruning good solutions.

Next, to minimize alignment problems related to synthetic bulges appearing with the *shape-from-silhoette* method, an iterative ICP algorithm is presented in Section IV. The corresponding experiments in Section VI-B show that the proposed method significantly reduces alignment error at the cost of increasing the computational cost appreciably. The algorithm was initially designed to use CAD models as references. These kinds of references do not have synthetic bulges, and the algorithm can work optimally, but reconstructed references can also take advantage of the algorithm. To have a CAD model of an object is common, for example, in a quality control context where a part has to be 3D scanned and checked against a model to detect geometric or surface errors.

Finally, a multicapture reconstruction approach is presented in Section V. This process refines object reconstruction with each new capture, reducing bulges on the object surfaces. This process leads to a more accurate reconstruction. The method's accuracy relies on the precise alignment of each reconstruction; if a CAD reference exists, the iterative ICP can be used to obtain the best results. On the other hand, the process can be performed incrementally; this can be useful to save memory usage in the computer implementation if needed.

As commented previously, this work was mainly developed in the research field of Industrial Inspection with a capture device in mind that uses a constellation of cameras to capture free-falling objects [11]. In Industrial Inspection, accuracy and speed are important, and the developed methods contribute to improving both requirements. The experiments were performed over synthetic data to validate the procedures, but the methods will be integrated in the real device, together with software and hardware optimizations for the more accurate but most time-consuming procedures.

Ultimately, we believe that while the initial motivation was specific, the techniques we propose have the potential to be beneficial in various contexts, contributing to more accurate 3D reconstructions and improved alignment.

REFERENCES

[1] A. Moncef and A. K. M'hamed, "A review on 3D reconstruction techniques from 2D images," in *Innovations in Smart Cities Applications*, 3rd ed. Cham, Switzerland: Springer, Feb. 2020, pp. 510–522.

[2] E. A. Hendriks, P. A. Redert, and Q. Wei, "Converting 2D to 3D: A survey," in *Information and Communication Theory Group (ICT)*. Delft, The Netherlands: Delft University of Technology, 2005.

[3] S. E. Hazzat, M. Merras, N. E. Akkad, A. Saaidi, and K. Satori, "Enhancement of sparse 3D reconstruction using a modified match propagation based on particle swarm optimization," *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14251–14276, 2019.

[4] M. Sarmah, A. Neelima, and H. R. Singh, "Survey of methods and principles in three-dimensional reconstruction from two-dimensional medical images," in *Visual Computing for Industry, Biomedicine, and Art*, vol. 6. Singapore: Springer, 2023.

[5] X. Fu and Y. Li, "A survey of image-based 3D reconstruction," in *Proc. Int. Conf. Mech., Electron., Control Autom. Eng. (MECAE)*. Amsterdam, The Netherlands: Atlantis Press, Mar. 2017, pp. 180–183.

[6] C. R. Dyer, "Volumetric scene reconstruction from multiple views," in *Foundations of Image Understanding*. Boston, MA, USA: Kluwer, 2001, pp. 469–489.

[7] K.-M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette across time part I: Theory and algorithms," *Int. J. Comput. Vis.*, vol. 62, no. 3, pp. 221–247, 2005.

[8] A. Y. Mulayim, U. Yilmaz, and V. Atalay, "Silhouette-based 3-D model reconstruction from multiple images," *IEEE Trans. Syst., Man, Cybern., B Cybern.*, vol. 33, no. 4, pp. 582–591, Aug. 2003.

[9] G. Haro, "Shape from silhouette consensus and photo-consistency," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4837–4841.

[10] J.-L. Landabaso, M. Pardàs, and J. R. Casas, "Shape from inconsistent silhouette," *Comput. Vis. Image Understand.*, vol. 112, no. 2, pp. 210–224, 2008.

[11] J.-C. Perez-Cortes, A. J. Perez, S. Saez-Barona, J. L. Guardiola, and I. Salvador, "A system for in-line 3D inspection without hidden surfaces," *Sensors*, vol. 18, no. 9, pp. 1–51, 2018.

[12] G.-J. Yoon, H. Cho, Y.-Y. Won, and S. M. Yoon, "Three-dimensional density estimation of flame captured from multiple cameras," *IEEE Access*, vol. 7, pp. 8876–8884, 2019.

[13] H. Kim, R. Sakamoto, I. Kitahara, N. Orman, T. Toriyama, and K. Kogure, "Compensated visual hull for defective segmentation and occlusion," in *Proc. 17th Int. Conf. Artif. Reality Telexistence (ICAT)*, Nov. 2007, pp. 210–217.

[14] W. Krajnik, Ł. Markiewicz, and R. Sitnik, "Segmented shape from silhouette reconstruction of the human body," *Sensors*, vol. 22, no. 3, p. 925, 2022.

[15] E. Simioni, F. Ratti, and L. Poletto, "Shape-from-silhouette for three-dimensional reconstruction from X-ray radiography," in *Proc. SPIE*, L. Pezzati and R. Salimbeni, Eds. vol. 8084, 2011, Art. no. 80840I.

[16] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 150–162, Feb. 1994.

[17] R. Szeliski, "Rapid octree construction from image sequences," *CVGIP, Image Understand.*, vol. 58, no. 1, pp. 23–32, 1993.

[18] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, Aug. 1987, pp. 163–169.

[19] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, Oct. 1994.

[20] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR04-004, 2004.

[21] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Proc. Symp. Geometry Process.*, 2013, pp. 113–123.

[22] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. Comput. Vis. ECCV*. Berlin, Germany: Springer, 2002, pp. 418–432.

[23] L. Wang and X. Sun, "Comparisons of iterative closest point algorithms," in *Ubiquitous Computing Application and Wireless Sensor*, J. J. J. H. Park, Y. Pan, H.-C. Chao, and G. Yi, Eds. Dordrecht, The Netherlands: Springer, 2015, pp. 649–655.

[24] Y. Gong, "Feature-based three-dimensional registration for repetitive geometry in machine vision," *J. Inf. Technol. Softw. Eng.*, vol. 6, no. 4, p. 184, 2016.

[25] S.-C. Lee and P. Bajcsy, "Automated feature-based alignment for 3D volume reconstruction of CLSM imagery," in *Proc. SPIE*, J. M. Reinhardt and J. P. W. Pluim, Eds. vol. 6144, 2006, Art. no. 61442Z.

[26] K. Fukunaga, *Statistical Pattern Recognition*, 2nd ed. New York, NY, USA: Academic Press, 1990.

[27] O. Del-Tejo-Catalá, J.-L. Guardiola, J. Pérez, D. M. Escrivá, A. J. Perez, and J.-C. Perez-Cortes, "Probabilistic pose estimation from multiple hypotheses," *IEEE Access*, vol. 11, pp. 64507–64517, 2023.

[28] M. Chaouch and A. Verroust-Blondet, "A novel method for alignment of 3D models," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, Jun. 2008, pp. 187–195.

[29] M. Kazhdan, "An approximate and efficient method for optimal rotation alignment of 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1221–1229, May 2007.

[30] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.

[31] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "Discrete-continuous optimization for large-scale structure from motion," in *Proc. CVPR*, Jun. 2011, pp. 3001–3008.

[32] K. Häming and G. Peters, "The structure-from-motion reconstruction pipeline—A survey with focus on short image sequences," *Kybernetika*, vol. 46, no. 5, pp. 926–937, 2010.

[33] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proc. IEEE Workshop Stereo Multi-Baseline Vis. (SMBV)*, Dec. 2001, pp. 131–140.

[34] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 9, nos. 1–2, pp. 1–148, 2015.

[35] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "DeepMVS: Learning multi-view stereopsis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2821–2830.

[36] O. Wiles and A. Zisserman, "SilNet: Single- and multi-view reconstruction by learning from silhouettes," in *Proc. Brit. Mach. Vis. Conf.*, 2017. [Online]. Available: https://arxiv.org/pdf/1711.07888v1

[37] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Modeling*, May 2001, pp. 145–152.

[38] R. Karinthi, "VII.6—Accurate Z-buffer rendering," in *Graphics Gems V*, A. W. Paeth, Ed. Boston, MA, USA: Academic Press, 1995, pp. 398–399.

[39] E. Swokowski, *Calculus With Analytic Geometry*. Boston, MA, USA: Prinfle, Weber and Schmidt, 1979.

ALBERTO J. PEREZ received the degree in computer engineering, in 1994, and the Ph.D. degree from the Universitat Politècnica de València, in 2005. In 1996, he was an Assistant Lecturer with the Universitat Politècnica de València. In parallel, he worked on I+D projects with the Instituto de Tecnología Informática (ITI), València. Currently, he is a full-time Lecturer with the Computer Science Department, Universitat Politècnica de València, and he continues his collaboration with the Instituto de Tecnología Informática, the Food Technology Department, Universitat Politècnica de València, and with the Medical Physics Department, Universitat de València. He has 14 articles in journals and ten articles in international congresses. He has been involved in more than 15 public and privately funded projects, being the technical manager in two of them. He participates in an international patent and a pending patent. His main research interests include computer vision, pattern recognition, 3D scanning, 3D manufacturing, industrial inspection, and food control.

JAVIER PEREZ-SOLER received the master's degree in intelligence systems, in 2012, and the Ph.D. degree in computer science engineering from the University Jaume I (UJI), Castellón, in 2017. He has more than six years of research experience in underwater robotics as part of the Interactive and Robotic Systems Laboratory (IRSLab), UJI, and as a Visitor to the Marine Australian Center for Field Robotics (ACFR), The University of Sydney. During this period, he was in charge of developing computer vision, deep learning, 3D simulations, and benchmarking solutions for autonomous underwater robots. Due to his research work, he has achieved six publications in high-impact journals, one book chapter, and more than 15 articles at important international conferences. Additionally, he has worked on two European financed projects (GUARDIANS and TRIDENT) and two Spanish government-funded projects (TRITON and MERBOTS). Since 2017, he has been a Software Engineer with the Research and Development Services Department, Instituto Tecnologica de Informatica Valencia (ITI), developing 3D modeling and reconstruction techniques to enhance quality assurance processes.

JUAN-CARLOS PEREZ-CORTES received the Ph.D. degree in computer science from the Universitat Politècnica de València. He is currently a Full Professor. He is also the Director of the Pattern Recognition And Image Analysis (PRAIA) Research Group, Instituto Tecnológico de Informática (ITI). He has led and coordinated research projects funded by public national and international entities in the field of medical imaging, industrial software, computer vision, pattern recognition, and free software. He teaches master's and Ph.D. courses in computer systems artificial vision and pattern recognition. He has published works in journals (15), three books, books, and conferences (17); and has been awarded by public and private entities.

JOSE-LUIS GUARDIOLA received the degree in industrial engineering, in computer engineering, and in bioelectronics specialists from the Universitat Politècnica de València, where he is currently pursuing the Ph.D. degree with the Electronic Department. He is currently a Technical Researcher with the Universitat Politècnica de València. He has been involved in more than ten public and private projects, leading some of them. He has published works in three journals and eight conferences. His main research interests include computer vision, pattern recognition, electronics, and signal processing.