3D Reconstruction of Small Objects by Photogrammetry: A Survey

Silvana Andrea Civiletto, Marco Demutti, Matteo Dicenzi, Vincenzo Di Pentima,
Elena Merlo, Emanuele Riccardo Rosi, Luca Tarasi, Simone Voto, Gerald Xhaferaj

Supervisors: Giorgio Ballestin, Fabio Solari

***Abstract*—3D reconstruction is a subject of great interest in several fields. In robotics, it is a key aspect of the robot's environmental awareness, being needed both for classification and manipulation of surrounding objects. In industrial settings, research & development, and rapid prototyping settings, it is often needed to have the 3D model of an object (e.g. a mechanical part) to design another part which is compatible (or a replacement). However, obtaining the 3D model is usually a time-consuming task which involves manual measurements and reproduction of the item using CAD software, which is not always feasible (e.g. for organic shapes). The necessity of quickly obtaining a dimensionally accurate 3D model of an object has led to the development of several reconstruction techniques, either vision based (with photogrammetry), using laser scanners, or a combination of the two. The contribution of this study is twofold. First, we provide a systematic review of techniques currently used for 3D reconstruction, which are split in three categories: approaches based on Shape-from-Silhouette, on Structure from Motion and on Deep learning. Then, we analyze the performances of currently available 3D reconstruction frameworks. Our results indicate that 3DF Zephyr represents a valid trade-off between the considered metrics, despite the small number of reconstructed points.**

I. Introduction

The problem of 3D reconstruction consists in determining the 3D shape of an object, given a finite number of measurements (which can be a combination of images, sensor information such as laser scanners, or a combination of more methods). In particular, two families of 3D reconstruction techniques exist: active methods, which actively interfere with the object to be reconstructed through radiometric or mechanical techniques (e.g. laser scanners, structured light) and passive methods, which only acquire images and/or videos in order to reconstruct the object [1].

In this paper we focus on passive 3D reconstruction problem and in particular on the RGB images based methods for small objects. Moreover, we consider only algorithms for full object reconstruction; therefore the single-view methods, which do not allow to completely estimate a 3D shape [2], are not discussed. Many algorithms have been developed during the years to solve this problem: it is not immediate to solve it, because of many factors: limited number of photos, image noise, calibration noise, numerical errors in the softwares, non-constant illumination in the scene, and many others. Moreover, it is a problem that requires long computational times in order to be solved, so it is very hard to extend it to real-time scenarios, even though some efforts have been made in this direction [3], [4], [5], [6].

The first objective of this paper is to give an overview of the various techniques for RGB camera based 3D reconstruction, which have been proposed in the recent years. These can be split in three categories: approaches based on Shape from Silhouette (SfS), on Structure from Motion (SfM) and on Deep Learning.

The SfS approach consists in performing an intersection among the silhouettes extruded from all the photos of the same object taken from different viewpoints; each extrusion can be imagined as a visual cone explaining the projection of a silhouette in its corresponding camera image. The intersection among all the visual cones is a volume named Visual Hull (VH), which approximates the volume of the object we are interested to reconstruct. In some cases other techniques are introduced in order to make the VH similar to our object, by exploiting some other information coming from illumination or pixel intensity values.

The SfM is another technique for estimating the 3D model of the object or the scene represented in a collection of images. It takes as input a set of unordered heterogeneous images and produces a dense point cloud representation. The entire SfM workflow can be divided into two macro phases: Structure from Motion (SfM), which computes a sparse point cloud and estimates the camera parameters, and Multi-View Stereo.

The last and most recently developed category of 3D reconstruction algorithms is based on Deep Learning: Convolutional Neural Networks (CNN) are employed in order to reconstruct the 3D shape of a single (multiple) object(s) given a set of images (single or multiple).

The second objective of the paper is to analyze the performances of currently available 3D reconstruction softwares. Five of these softwares (3DZephyr, Colmap, Meshroom, Metashape and Regard3D) are tested on synthetic datasets, and their quality is evaluated according to some criteria (such as reconstruction time, mean error, visual inspection of the reconstruction results, documentation).

The rest of this paper is organized as follows: in section II an overview of the main SfS methods used to compute VH is given. They are divided in three groups: Volumetric approaches are presented in II-A, Surface approaches in II-B and Hybrid approaches in II-C.

Sections III-A and III-B describe the methods and the algorithms considered respectively in SfM and MVS.

In Section IV, a brief overview of the Deep Learning approach and a possible taxonomy of the algorithms are given. In particular some of the most recent networks are studied and categorized.

In section V, the experimental setup of the performance analysis is described.

Finally, in sections VI and VII the discussion of the performed analysis and the conclusions of the overall survey are provided.

II. Shape from Silhouette (SfS)

The SfS approaches are based on the contour of the 2D projections of the object, also called silhouette.

The first step consists in the segmentation of all the images, in order to obtain the silhouettes. To do this, various techniques can be employed, as reviewed in [7]. The most important and simplest ones are the Running Gaussian average and the Temporal median filter methods.

The second step is the computation of the visual cones intersection obtained by back-projecting, in the 3D space, the lines passing through the camera center and the points on the silhouette, for each image. The result is called Visual Hull (VH) [8], which is by definition the maximal solid shape consistent with the object silhouettes [9], and therefore it is an upper bound of the object. The VH does not model object concavities, which cannot be derived from the silhouettes; therefore, to obtain a more accurate object representation, the VH is often refined by using other methods, usually based on photo-consistency [10], [11], [12], [13], [5], [14] or on shadow information [15]. As shown in Fig. 1, there are at least three main schools for computing the VH: volumetric, surface-based and hybrid.

A. Volumetric Methods

The most common and oldest approach for computing the VH is the volumetric one, used already in [16], where the VH is computed as a volume. There are many possibilities to compute it, the most diffused corresponds to voxel-based approaches, where the 3D space is discretized into a regular grid. Additionally, alternative methods consists in Marching Intersections in which the viewing cones intersection is estimated using an optimized data structure, Polyhedra solutions based on geometric procedures and Image-Based ones relying on the image space concept.

Volumetric methods can be split in two main categories: *voxel-based* and *non voxel based* methods. The *voxel-based* approaches employ a discretization of a portion of the 3D space into cubes of chosen dimension (also called resolution), named voxels. This process is called voxelization. In the voxel-based methods, usually all the voxels are projected on all the available silhouettes: the basic idea is that voxels whose projection is out of at least one silhouette are not part of the VH and thus removed. A popular tree structure that allows to do this in an optimized way is the octree, introduced in [17] and used in subsequent papers, like [3], [18], [19], [20].

A critical aspect is related to the assumption of having perfect silhouettes, as done in many papers [4] [5] [21] [22]. Since this is not true, e.g. because of some inconsistencies generated by limitations in the Hardware, many approaches [23] [24] [13] [6] [25] [26] [27] devised solutions that try to deal with this issue, implying a reduction of the Voxel Missclassification due to 2D errors (segmentation, calibration, etc).

A classical example can be the SPOT algorithm [6], which acts on the Reprojection Test. Basing on the number of pixels defining the condition of belonging we can distinguish: Single Pixel, Complete and Sampled Reprojection tests. The first one, used in [26] [28], compares to the silhouettes only the pixel corresponding to the voxel center projection in all the views. In the second instead all the pixels belonging to the voxel splat are considered. For the Sampled Reprojection solution proposed by Kanade et al. the minimum number of pixels is defined depending on the computed error probability, to achieve a desired value for it. Obviously it is a part of a trade-off in terms of efficiency: a lower probability corresponds to a larger computation time and vice versa. This allows to obtain a simple and fast approach, designed with the purpose of working in real time (from video frames). Improving this initial idea, Landabaso [27] proposes the concept of "Unbiased Hull", that corresponds to the volume of the real object discarded in the standard SfS processing. In fact, the classical intersection implies an unbalance between false positives and false negatives.

To overcome this limitation a second step, in which the VH is re-evaluated, is performed in order to recover the VH loss. Consequently the desired balance (highest F-score) is achieved, since the total error improves with an increasing recall but losing in terms of precision, because of more false positives.

Another method that deals with this issue is based on the Dempster−Shafer theory [26]. In particular instead of projecting the voxel center on the image planes establishing whether the point is in or out the silhouette, an evidence (weaker than probability) is computed taking into account some information about the setup (relative angle between two cameras). If the evidence of being inside the silhouette is bigger, then the voxel will be considered occupied and vice versa. The proposed method has been compared to the standard SfS and the Shape from Inconsistent Silhouette (SfIS) [27], and the results show that it obtains better reconstructions, especially under noisy conditions.

Some non-probabilistic methods for dealing with the problem of non-ideal silhouettes have been proposed as well, for instance those based on the minimization of silhouette inconsistency.

An example is the Shape from Inconsistent Silhouette with Reprojection of Error (SfISRE) [23], where a cost function, which takes the reprojection error into account, is minimized to obtain the estimated volume. Two versions of the cost

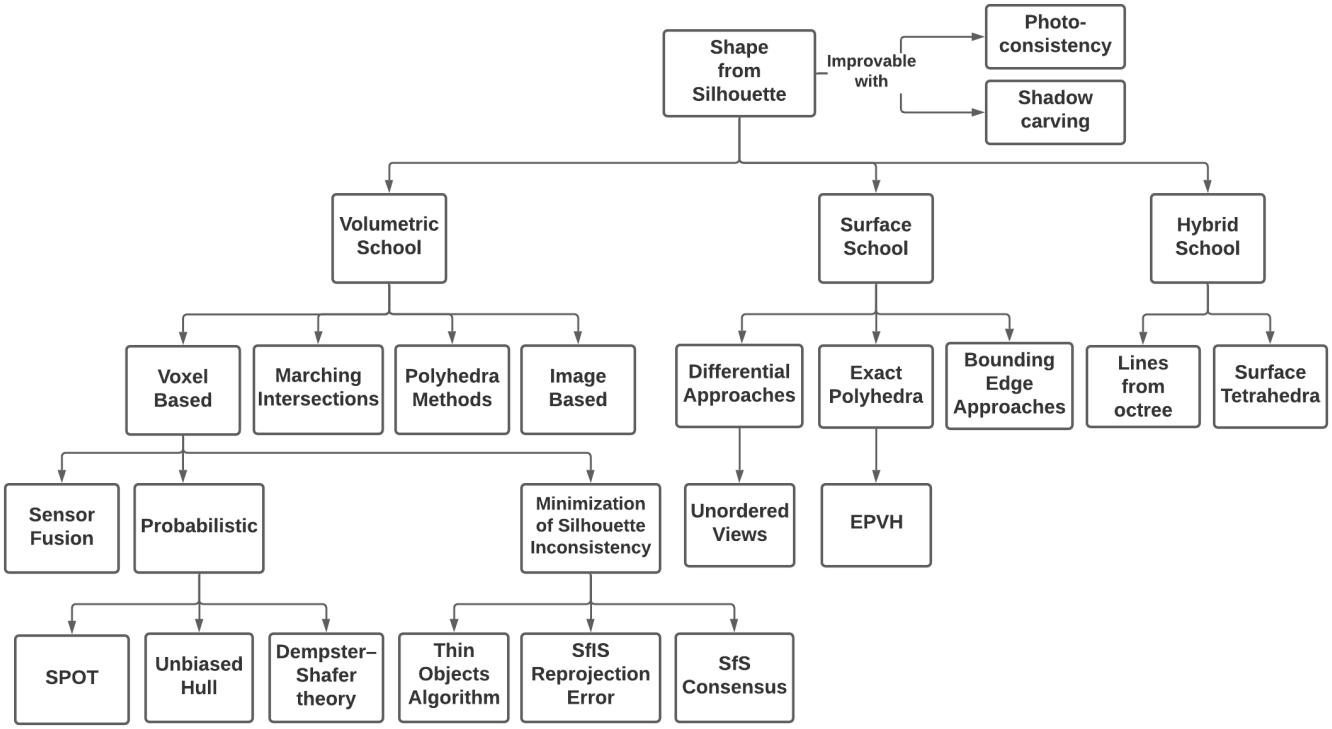

Fig. 1: SfS approaches divided according to: VH definition (volume, surface and hybrid), general technique to compute VH and specific algorithms. If no further arrow is used for a category, the found solutions are not different enough to be relevant. Additionally, SfS refinement methods are indicated in the top right corner.

function are proposed: one for binary silhouettes and another for the ones given in probability maps format, where instead continuous values $\in [0, 1]$ (interpretable as probabilities of a pixel to belong to the silhouette) are used.

A very similar approach is proposed in [25], where the cost function is simpler and usable, differently from [23], for both SfIS and Shape from Probability Maps (SfPM) inputs. It is able to achieve better performances, compared to [6], with thin objects, the most critical scenario for 2D error.

In [24] the author proposes the SfS Consensus (SfSC), an approach based on variational calculus; within this algorithm, all the silhouettes contribute to decide whether each voxel is inside or outside the VH, according to a functional that is minimized with an iterative procedure: by doing so, the contribution of silhouettes with missing parts is mitigated and the number of false negatives is reduced. Here, as in SfISRE, a regularization term is employed in the minimization: this allows to select the solution with minimal surface variation, but it does not necessarily guarantee to obtain a shape that is faithful to the original one; also, no refinement method for the VH is considered.

The tests in [23], [24] are performed on sets of silhouettes where two or more are corrupted with missing parts. Among SfISRE and SfSC, the first one usually has a worse recall because the cost function also has a sparsity term, proportional to the number of points occupied by the VH; this often yields thinner volumes than the actual one. On the other hand, SfSC has a worse precision and better recall, since often the estimated volume is slightly larger than the true one.

Even in the case where not all silhouettes are corrupted, SfISRE and SfSC are found to have a better F-measure (that is, global quality of the result) than simply computing the VH by simple intersection of the visual cones associated to the non-corrupted silhouettes. Indeed, the latter approach avoids missing parts in the resulting VH but the shape is overestimated due to the utilization of less silhouettes, a problem not found in SfISRE and SfSC. In [13], photo-consistency is used for VH refinement, obtaining better results in the performed tests.

Non voxel-based methods, on the other hand, do not make use of voxels to compute the VH. One noteworthy example is the Image-Based VH (IBVH) approach [4] [5], which is based on selecting one of the views, called desired view, and back-projecting in 3D space all rays passing through the corresponding camera center and the pixels of the desired image. Then, such rays are projected on all the other images, and the segments where the projected rays intersect each silhouette are computed. The segments are subsequently lifted back in the 3D space, and the intersection of all the lifted segments belongs to the VH. This algorithm is optimized for real-time employment, but produces a view-dependent result. In [5] the Photo Hull, which is defined as the spatially largest

set of points in 3D space that project to photo-consistent colors in the reference images, is computed by integrating the Image-Based VH method with a photo-consistency criterion.

Another method to compute the VH not relying on voxels is the one based on Marching Intersections (MI) [29], a data structure used to represent 3D shapes. In particular, three perpendicular directions are chosen and, for each direction, an array of rays is intersected with the solid; the points where the solid intersects each ray are stored to encode the shape. The procedure to convert the solid to an MI structure automatically removes high-frequency details: this can be functional in the case of noisy silhouettes. The intersection is then computed by logic AND operations in an efficient way.

An alternative non-voxel-based approach is the centripetal pentahedron model [21],[22], based on the idea that the space can be partitioned with a set of infinite triangular pyramids; the pyramids are then cut into a set of pentahedrons, obtained by efficient 2D intersections of their edges on all the silhouettes. The set of pentahedrons obtained at the end of the procedure constitute the VH.

B. Surface Methods

These methods consider the VH as a surface and can offer some advantages over the volumetric voxel-based methods, which are plagued by discretization errors and can become computationally heavy as the resolution increases. In [9], for example, the classical visual cones intersection is revisited in order to progressively compute geometrical primitives (edges, frontier points, faces) leading to the complete object surface. It guarantees a polyhedrical pixel-exact surface, with topological properties of manifoldness and watertightness. In addition to that, it is temporally efficient, in fact, compared to [30] where a similar solution is designed, it is able to achieve the same quality but with a lower number of primitives. Nevertheless it is affected by the (typical) numerical instabilities related to degenerate configurations.

In [31], the point-tangent plane duality in homogeneous coordinates is exploited to compute surface points. The successive usage of a 2D Delaunay triangulation and a comparison with the original silhouette, allow to build the tetrahedrons composing the object surface. This solution is interesting because it solves surface school typical issues, such as the absence of robustness and the impossibility of working with unordered views, while maintaining its accuracy, better than volumetric approaches even using high resolution for the voxels. Moreover, it also guarantees to respect original surface properties, more than other approaches from the same school [9].

Another solution for finding surface points is using bounding edges, that are segments obtained by back-projecting viewing lines in the 3D space, projecting them in each image, computing the intersections with each silhouette and re-projecting the result in the 3D space: the overall intersection of such re-projections is taken. For the reconstruction, we use the fact that each bounding edge has at least one contact point on the surface; such point is found by employing some techniques such as the photo-consistency. For example in [12] the idea is that of having a lot of silhouettes for improving the quality of the final VH, but using a limited number of cameras, by exploiting information about pixels intensity. The object moves arbitrarily for a certain time period and each camera takes photos of it for all the time instants. So if the cameras are K, while the instants are J, $K \times J$ silhouettes are obtained. Thanks to photo-consistency, Cheung et al. are able to estimate the object motion and perform an alignment among the various shots, such that all the images are considered as being captured at the same instant.

An ulterior technique that uses the contact points extracted from bounding edges is that of performing stereo matching in the two views considered as the best ones for the currently examined point [14]. Once the best viewable stereo image pair is obtained, the contact point on any bounding edge can be identified by template matching and the epipolar constraint. Once the contact point is found, a sphere centered in it is created and projected on the selected image pair: in this way many other correspondences are found by reducing the searching range.

C. Hybrid Methods

Hybrid methods combine the advantages of volumetric and surface approaches, trying to overcome the disadvantages of both; for instance [20] initializes the VH with a voxelisation octree-based technique. Then it is refined by first converting the volume to a line-based model, later reprojecting each 3D line on all images. Finally, the intervals where the reprojected lines intersect the silhouettes are lifted back in 3D space, and then all the lifted segments are interescted: only the result of the intersection are kept in the VH. Better results than pure voxelizations are obtained in tests.

Another example of this category can be found in [28] where, from the initially individuated surface points, the tetrahedrons composing the object are extracted. The points estimation is based on the intersection of viewing lines, obtained by considering a selection of points from silhouettes contours. Then, following the 3D Delaunay triangulation, the solids are compared to the silhouettes to decide whether they belong or not to the object.

This approach succeeds in contemporary achieving accuracy, typical of the surface school solution, and efficiency, belonging to the volumetric one. On the other hand, it introduces a new drawback related to the dependency of the performances with respect to the ratio between the number of images and the number of silhouettes contour points.

To conclude, Table I summarizes the main aspects of the studied papers, offering the reader a fast comparison and simplifying his research.

III. Structure from Motion and Multi-View Stereo Photogrammetry

Traditional stereophotogrammetry methods allow to reconstruct 3D representations of scenes and objects based on the binocular changing vision of an object, that is either moving or

TABLE I. Table with the notable SfS papers. A tick means Yes; a cross means No. Meaning of the columns: `Study` contains the reference to the paper; `VH method` contains the method used to compute the VH; `Calibration`indicates whether calibration is necessary or not; `VH refin`indicates whether the VH is refined in the algorithm; `Texture` indicates whether texturisation is present; `Camera views` contains the number of camera views used in the tests; `Time indicated` specifies whether test timings are shown in the paper; `Error` contains the measure of error used in the paper.

Study	VH Method	Calibration	VH Refinement	Texture	Camera Views	Time indicated	Error
[10]	Volumetric	✗	✓	✗	36	✗	Points distance
[11]	Voxelization	✓	✓	✓	12, 18	✗	Visual plots
[32]	Bounding edge	✓	✓	✗	8	✗	Mean Absolute Deviation
[12]	Voxelization	✓	✓	✗	90, 132	✓	Visual plots
[6]	Probabilistic	✓	✗	✗	5	✓	Visual plots
[25]	MSI	✓	✗	✗	30	✓	Visual plots
[27]	Probabilistic	✓	✗	✗	5	✗	Precision, Recall, F-score
[29]	Marching intersections	✗	✗	✗	36	✗	Points distance
[14]	Voxelization	✓	✓	✓	72	✓	Visual plots
[15]	Voxelization	✓	✓	✓	24, 72	✗	Visual plots
[33]	Volumetric	✓	✗	✓	36	✗	Visual plots
[26]	Probabilistic	✓	✗	✗	6, 8	✓	Precision, Recall, F-score
[23]	MSI	✓	✗	✗	72	✗	Hausdorff distance, PRF
[24]	MSI	✓	✗	✗	72	✗	Hausdorff distance, PRF
[13]	MSI	✓	✓	✗	72	✗	Hausdorff distance, PRF
[4]	IBVH	✗	✗	✓	4	✗	Visual plots
[5]	IBVH	✓	✗	✓	4	✗	Visual plots
[34]	Volumetric	✓	✗	✓	4	✗	Num. of missing/extra pixels
[9]	Surface	✓	✗	✗	3:42	✓	Contour, vertices, faces
[31]	Surface	✓	✗	✗	36, 20	✗	Avg point-to-surface distance
[28]	Hybrid	✓	✗	✗	4:42	✓	Visual Plots
[35]	Volumetric	✓	✗	✗	10	✗	Silhouette calibration ratio
[36]	Volumetric	✓	✗	✗	8	✗	Visual plots
[20]	Hybrid	✓	✗	✗	33, 36	✗	Hendrik error
[37]	Hybrid	✓	✗	✗	33, 36	✗	Hendrik error
[21]	Polyhedra	✓	✗	✗	36	✓	Number of data structures
[22]	Polyhedra	✓	✗	✗	36,70	✓	Number of data structures
[19]	Voxelization	✓	✗	✗	4:8	✓	Mean projection error

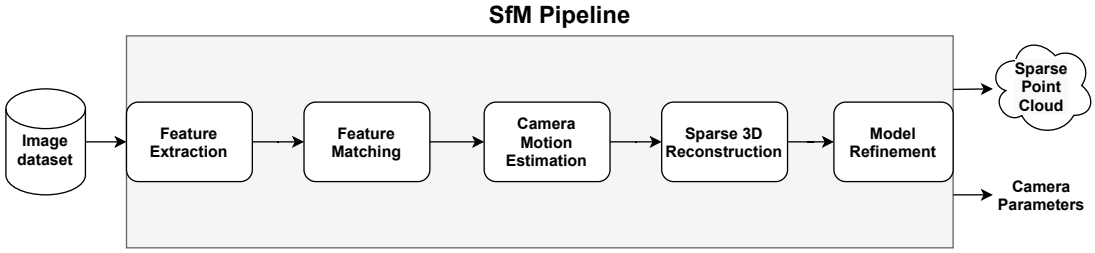

Fig. 2: Given a set of uncalibrated 2D images, the main stages of a generic SfM pipeline are: feature detection, feature matching, camera motion estimation, sparse 3D reconstruction, model refinement. The process produces: the camera parameters of each image and a 3D point cloud.

observed from a moving point. This section describes one of the most popular framework, which combines Structure from Motion (SfM) and Multi-View Stereo (MVS) techniques to obtain a dense 3D model starting from a set of unordered images.

A. Structure from Motion (SfM)

Structure from Motion is a technique for estimating 3D models from a sequence of unordered overlapping 2D images. It has been widely used because it allows to deal with unordered sets of heterogeneous images. Sometimes "SfM photogrammetry" is used to define the entire 3D recontruction workflow, up to the creation of a dense point cloud. However, rigorously speaking, the term refers to the specific process that provides camera parameters and a sparse point cloud. As depicted in Fig. 2, it consists of five phases [38], [39]:

1) Feature extraction: it is a process that aims at identifying and extracting a set of interesting points (i.e. keypoints) from a data set of n images, in terms of location (detectors) and description (descriptors). One of the most commonly used detectors is SIFT [40] [41], which is invariant to image changes in the scale or rotation and also robust to illumination changes. Other detectors considered in the literature are AKAZE [42] and SURF [43].

2) Feature matching: it follows the feature extraction process and consists of matching the feature points of each image pair in the data set. Approximate Nearest Neighbors (ANN) and RANdom SAmple Consensus (RANSAC) are some of the common methods. The first performs the matching minimizing the Euclidean distance of feature point descriptors across two images [44]. The latter removes false matches, by performing the random sampling of matched features iteratively to estimate the epipolar geometry [45]. Other methods are FLANN [46] and ORSA [47]. The computational complexity of exhaustively matching grows quadratically with the number of images in the data set. For this reason, one may consider global similarity measures such as vocabulary trees [48], [49] which succeed in reducing the overall complexity.

3) Camera motion estimation: starting from the feature points of each image pairs, it computes the intrinsic (focal length and radial distortion) and extrinsic (position and orientation) camera parameters. In this regard, it is necessary to estimate the Essential matrix or the Fundamental matrix, which both allow describing the correspondence between two matching features. The first contains the extrinsic parameter only and can be estimated with the Five-point algorithm [50], which is based on five matched feature points. The latter, the Fundamental matrix, contains both the extrinsic and intrinsic camera parameters and can be estimated by using the Eight-point algorithm, which works on a set of eight matched feature points [51]. Alternatively, the camera parameters can be computed with the DLT algorithm, which is based on least square methods and needs 3D coordinates of points along with feature points of pixel coordinates [52].

4) Sparse 3D reconstruction: it computes the 3D coordinates of each point corresponding to the matched features through the Triangulation algorithm [44], [53]. Thus, a 3D sparse point cloud is originated.

5) Model refinement: it is the last step of SfM. The overall process is very sensitive to the accuracy of the camera models estimation, since they allow to restrict the 2D matching problem to a 1D matching problem. As pointed out in [54], this estimation phase may suffer from numerical instability, therefore a correction is needed to avoid SfM drifting. Bundle Adjustment (BA) [55] is a nonlinear least-squares algorithm used to refine both the camera parameters of each image and the location of 3D points, minimizing the reprojection error. Overall, BA increases the robustness of the solutions [56].

Table II summarizes the most common techniques and algorithms considered in each step of the SfM pipeline.

A further classification of SfM techniques can be made, depending on the way the pipeline is carried out. *Incremental SfM* is the most common approach. The model is initially seeded with a carefully selected initial image pair. Then, 3D reconstruction is performed by repeatedly adding matched images, triangulating features, and refining the model performing bundle-adjustment. The complexity is $O(n^4)$ given *n* images, because of the repetition of the bundle adjustment phase [48]. Wu [57] proposes an improvement of the classic method, based on a novel BA strategy, showing that incremental SfM can require only $O(n)$ time on many major steps. The main disadvantage of this approach lies in the drift due to the accumulation of errors. Also, the quality of the reconstruction is highly sensitive to the choice of the initial image pair [58]. *Global SfM*, on the contrary, aims at performing the reconstruction of all the images at once. In particular, most methods first compute the global camera rotations, then the camera translations and the 3D structure. The main advantage is the reduction of drift since residual errors are distributed evenly [58]. However, global approaches may suffer from inaccurate pairwise geometries. Also, since the minimization depends on the reprojection errors, space and time requirements may get very large. In addition to the two mentioned above, other less common approaches exist in the literature, such as Hierarchical SfM [59].

B. Multi-View Stereo (MVS)

Once the SfM step in the overall 3D reconstruction is completed, the intrinsic and extrinsic camera parameters are known and a sparse point cloud model has been computed. However, most studies do not consider the sparse point cloud to be the final result. It is often necessary to enhance the density of the point cloud coming from the SfM, through dense image matching algorithms. In this regard, the Multi-View Stereo (MVS) technique is often employed to derive a dense 3D model of the object/scene from a set of calibrated images, by using stereo correspondence. The literature proposes four main categories of MVS algorithms, as shown in Fig. 3, where each of them differs from the others depending on the model considered in the process [63]:

1) Patch-based: the object or scene is represented through a collection of patches, where each patch is a three-dimensional rectangle whose orientation is set so that one of its edges is parallel to the x-axis of the reference camera. Among them, Patch-based MVS (PMVS) is an algorithm that works considering such patches, developed by Furukawa et al. [64]. As pointed out by the authors of the article, the algorithm takes as input the camera calibration parameters and the sparse 3D reconstruction coming from the SfM module, and returns a dense set of rectangular patches that cover completely the surface of the object. The algorithm consists of three phases: matching, expansion and filtering. In the first phase the goal is to generate a sparse set of patches. This is done at first by detecting interesting features (i.e. edges and corners) through Harris Corner Detector and Difference-of-Gaussian (DoG), and then by matching such features among multiple images through local photometric consistency with Normalized Cross Correlation (CNN). Then, the set of patches is extended to the nearby pixels, obtaining a denser set. Finally, the filtering phase aims at reducing the noise introduced in the previous step, by removing the incorrect matches. As highlighted in [65], the PMVS algorithm may be potentially complex in some situation (i.e. global reconstruction). To overcome this issue, one may refer to clustering techniques, which allow the MVS algorithm to be executed in parallel for each cluster available. Clustering Multi-View Stereo (CMVS) is an algorithm that

TABLE II. SfM framework and methods

Step	Algorithms and methods
Feature extraction	SIFT [40], [41], ASIFT [47], AKAZE [60], SURF [61], Harris corner detector [53]
Feature matching	RANSAC [45], ORSA [47], ANN [44]
Camera motion estimation	DLT [52], Five-point algorithm [52], Eight-point algorithm [62]
Sparse 3D reconstruction	Triangulation [44], [53]
Model parameters correction	Bundle adjustment (BA) [55]

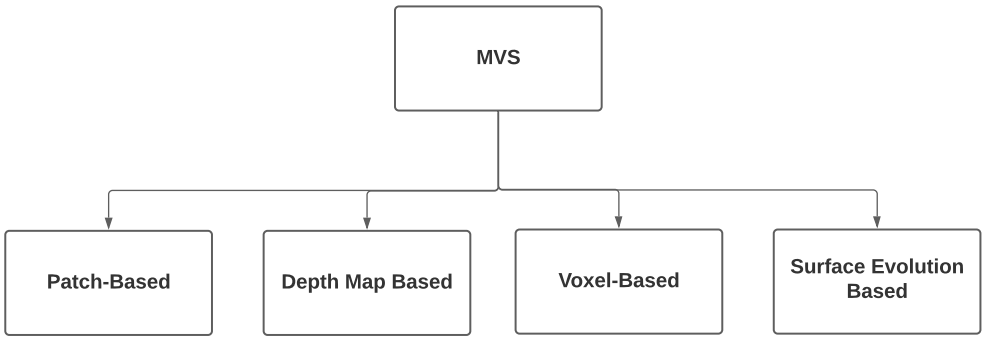

Fig. 3: MVS cathegorization, according to the scene representation during the pipeline [63]: patch-based, depth map based, voxel-based, surface evolution based

belongs to the previous mentioned technique [64], and allows the system to process a larger number of images at run time [65].

PMVS2 is software that reconstructs a realistic and detailed model based on PSVM [66]. It has the advantage of preserving only the rigid structures and it can deal with possible occlusions (i.e a person walking in the scene). Besides, the algorithm is also robust to different environmental conditions (like illumination). However, its robustness to errors in the camera parameters may not be true in general: it can degrade if, for example, the matching between features is performed with a distance larger than two pixels [64].

2) Depth map based: the second category of MVS is based on a set of depth maps, which are computed with image-space methods [67]. More specifically, this algorithm computes one depth map for each image in the data set and then it merges all the depth maps together. According to [68], this kind of approach is suited whenever there is the need of scaling to large scenes. In addition, the computation of the depth maps is performed in parallel [69]. On the other hand, the depth map computation is generally noisy, highly redundant and computationally complex [68], [69].

3) Voxel-based: the third category of MVS is based on Voxels, which are three-dimensional grids. This kind of methods have performances constrained to the choice of the voxel grid resolution [70]. Moreover, such choice depends on the object/scene modelled, so it is necessary to know in advance the object shape to properly select the voxel resolution and obtain good performances [64].

4) Surface evolution based: the last category of MVS is based on deformable polygonal meshes or on voxels. These algorithms includes an iterative minimization process of the a given cost function [67].

Table III summarizes some of the available software that allow the user to perform SfM and MVS. Note the distinction between open-source (tick) and not open-source (cross) software.

TABLE III. Software for dense 3D reconstruction

Software	SfM	MVS	Open-source
COLMAP [40]	✓	✓	✓
PMVS2 [69]	✗	✓	✓
Agisoft Metashape [71]	✓	✓	✗
Bundler [72]	✓	✗	✓
Pix4DMapper [73]	✓	✓	✗
VisualSfM [74]	✓	✓	✗

It is worth mentioning that Multi-View Stereo (MVS) may not be the last step in the SfM+MVS 3D reconstruction: one may want to convert the MVS output (i.e. the dense point cloud) to a mesh model. This is achieved through Poisson Surface Reconstruction [75], or with different techniques [76], [77]. For a deeper understanding about such techniques, the reader may refer to Furukawa et al. [78].

IV. Deep Learning

In the recent years 3D reconstruction using convolutional neural networks (CNN) has been focus of research and has overcome state-of-the-art methods in terms of performance. The classical methods (Structure from Motion, Shape-from-X)

solved the problem by a geometric perspective, focusing on formalizing mathematically the 3D to 2D projection process. The solutions typically require multiple images captured using calibrated cameras. However this may not be feasible in many situations.

The avenue of deep learning techniques and the increasing availability of large training datasets have led to a new generation of methods that are able to recover the 3D geometry and structure of objects from single or multiple RGB images without the complex camera calibration process. Following the definitions used in [79], we start recalling how a 3D reconstruction problem can be defined in the context of CNN's, then in Table IV we categorize the networks (manly the recent ones), in terms of the input, the output, their description model and the training procedure.

Han et al. [79] defined the problem as the following: let $I = \{I_k;\ k = 1, ..., n\}$ be a set of $n \geq 1$ RGB images of one or multiple objects X. 3D reconstruction can be summarized as the process of learning a predictor f_θ that can infer a shape $\hat{X}$ that is as close as possible to the unknown shape X. In other words, the function f_θ is the one that minimizes a loss function $L(I) = d(f_\theta(I), X)$. Here, θ is the set of parameters of f and $d(-,-)$ is a certain measure of distance between the target shape X and the reconstructed shape $\hat{X}$.

Some examples of $d(-,-)$ can be the Mean Squared Error (MSE) [80] or the Intersection Over Union (IoU) [80], [81], [82]. In addition, the accuracy can be also measured by Mean of Cross Entropy (CE) [80], [83], by Earth Mover Distance (EMD) [82] or by Chamfer Distance (CD) [82].

I can be a single image [84], [82], [85], [86], [80], [87] or multiple images [88], [89], [84], [83], [90], [91], [81], [85] captured by a camera whose intrinsic and extrinsic parameters are known or unknown.

When there is only one input the problem becomes challenging due to the ambiguity in the 3D reconstruction. Moreover the input can be depicting one or multiple 3D objects belonging to known or unknown shape categories. It can also include additional information such as silhouettes, segmentation masks and semantic labels as priors to guide the reconstruction.

There are different representations for the output. Each of them present some pros and cons. Moreover they add constraints in the design of the network. *Volumetric representations* are based on voxel grids [84], [81], [85], [91]. They are very expensive in terms of memory requirements when an high resolution is needed.

Surface-based representations are based on meshes and point clouds [83], [82], [80], [87], [86]. They are memory efficient, but they are not regular structures, hence don't easily fit into deep learning architectures.

Other representations utilize depth maps instead [90], [88], [89]. These representations require always an additional fusion step to retrieve the shape of the scene. For instance Yao et al. [88] use a visibility-based fusion algorithm to retrieve a point cloud representation [92].

Deep learning approaches can be categorized based on the underlying architecture. In general, the architectures are composed by an encoder h followed by a decoder g, $f = g\ o\ h$. The encoder maps the input into a latent variable x, referred as feature vector, using a sequence of convolutions and pooling operations, followed by fully connected layers. The decoder decodes the feature vector into the desired output by using either fully connected layers or a deconvolution network (sequence of convolution and upsampling operations). The former is suitable for 3D point clouds, while the latter is used to reconstruct volumetric grids or parametrized surfaces. Naturally the above representation is still abstract because additional blocks are used in the literature.

3D reconstruction algorithms also depend on the availability of large training datasets. *Supervised* techniques require images and their corresponding 3D annotations in the form of: full 3D models represented as volumetric grids, triangular meshes, or point clouds [84], [86], [91], [82]; depth maps [88], [89], [83], [90], which can be dense or sparse. *Weakly supervised* and *unsupervised* techniques rely on additional supervisory signals such as the extrinsic and intrinsic camera parameters and segmentation masks [85], [87].

The main challenge in collecting training datasets for deep learning-based 3D reconstruction is two-fold. First, while one can easily collect 2D images, obtaining their corresponding 3D ground truth is challenging. Second, datasets such as ShapeNet [93], which is one of the largest 3D datasets currently available, contain 3D CAD models without their corresponding natural images. This issue has been addressed in the literature by data augmentation, which is the process of augmenting the original sets with synthetically-generated data. For instance, one can generate new images and new 3D models by applying geometric transformations, e.g., translation, rotation, to the existing ones. One can also render, from existing 3D models, new 2D and 2.5D (i.e., depth, silhouettes) views from various viewpoints, poses, lighting conditions, and backgrounds.

V. Commercial Software Comparison

As part of this study, we evaluated the performance of five currently available 3D reconstruction softwares. Namely, we tested two commercial softwares (3DF Zephyr [94] and Agisoft Metashape [95]) and three open-source softwares that presented adequate documentation (Colmap [96], Meshroom [97] and Regard3D [98]). Each software is based on SfM approach except for 3DF Zephyr, which is implementing SfS. Tests were carried out on a Intel i7-6700HQ processor, NVIDIA GeForce GTX 960m graphics card and Windows 10 operating system.

We decided to use a synthetic dataset, IVL-SYNTHSFM-v2 [99], the benefits of which are presented in the following. It contains five objects: bicycle, empire vase, hydrant, jeep and statue. For each object, 8 scenes with different combinations of lighting, depth of field and motion blur are created. The images are taken with a perspective virtual camera from 100 points of view. Data also includes information about camera intrinsic and extrinsic parameters for each image, as well as the ground truth geometry of the 3D models, provided as a *.obj* files. The images are rendered with Blender. In our tests

TABLE IV. In the following table we have summarized the characteristics of the networks that have been investigated in this study.

Method / Year	Input	Output	Training	Network Architectures
MSVNet / 2018 [88]	multi-view	depth maps	supervised, depth maps	2D CNN + Cost Volume + Refiner
R-MSVNet / 2019 [89]	multi-view	depth maps	supervised, depth maps	2D CNN + GRU
Pix2Vox-F / 2019 [84]	single/multi-view	voxel grid	supervised, voxel grid	Encoder + Decoder
Pix2Vox-A / 2019 [84]	single/multi-view	voxel grid	supervised, voxel grid	Encoder + Decoder + Refiner
Atlas / 2020 [91]	multi-view (calibrated)	voxel grid	supervised, TSDF	2D CNN + Encoder + Decoder
DeepMVS / 2018 [83]	multi-view	point cloud	supervised, depth maps	Encoder + Decoder
DISN / 2019 [82]	single-view	mesh	supervised, SDF	Encoder + GCN decoder
DeepSfM / 2020 [90]	multi-view	depth maps	supervised, depth maps	Encoder + 3D CNN + Refiner
McRecon / 2017 [85]	single/multi-view	voxel grid	weakly supervised, 2D masks	Encoder + Decoder + Discriminator
Map Prediction Net / 2019 [80]	single-view	point cloud	supervised, 2D masks	Encoder + 2D decoder + Fourier transform
SoftRas / 2019 [87]	single-view	mesh	3D unsupervised	Encoder + FC decoder + Renderer
Mesh R-CNN / 2020 [86]	single-view	mesh	supervised, mesh	Predictor + Refiner

we used two of the provided objects (empire vase and statue), both in two different scenes, fixed sun lighting (fs) and moving sun lighting (ms). Our main goal was to test the performance of the different softwares, but also to understand if there was a difference in the reconstruction of different objects and scenes.

Each software was used with its default parameters. The resulting reconstructed dense point clouds were exported as *.ply* files and imported in CloudCompare for the comparison with the ground truth. Since this latter is provided as a 3D model, on CloudCompare we could consider it as a mesh or a point cloud made of mesh vertices. So, we decided to consider two measures of error between the reconstructed and the ground truth models: absolute cloud-to-cloud (C2C) and cloud-to-mesh (C2M) distances. While the first one takes the vertices of the ground truth mesh as reference, the other one uses the meshes themselves. Both measures are computed by CloudCompare as the Euclidean distances between each point of the reconstructed point cloud and its nearest neighbor of the truth model (cloud or mesh respectively).

Before computing the error between the reconstructed model and the ground truth, it was necessary to perform the alignment of the two on CloudCompare. In particular, we segmented the reconstructed point cloud, discarding the artifacts not belonging to the model (e.g. the floor). Then, the reconstructed model was scaled in order to match the dimensions of the ground truth model. A coarse, manual alignment was done in order to proceed next with a fine alignment, done automatically through the Iterative Closest Point (ICP) algorithm. Finally, it was possible to compute the distances between the two models. The ground truth of both objects is shown in Fig. 4, along with an example of reconstructed model. The comparison is shown in Table V, taking into account the distances from the ground truth, the time necessary to complete the reconstruction and the number of reconstructed points. The images of the reconstructed objects are shown in Fig. 5 and 6. The color scale represents the distance of each point from the ground truth.

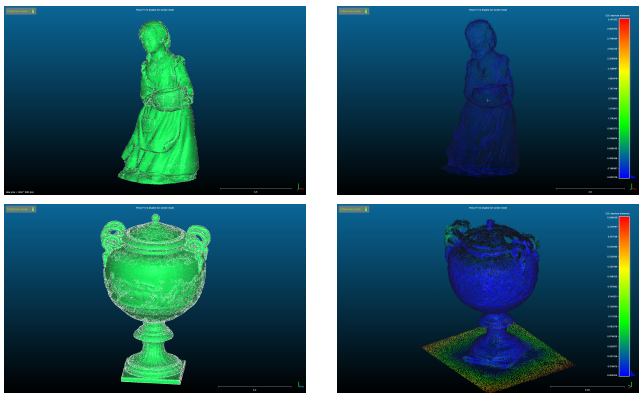

Fig. 4: On the left: the ground truth for the statue and vase object. On the right: a point cloud obtained through Colmap. The displayed color scale indicates the Euclidean distance between each point of the point cloud and the nearest point of the ground truth point cloud or mesh, which we use as error measure. The blue points represent the least amount of reconstruction error.

VI. Discussion

A. Software comparison analysis

Given the results shown in Table V and in Fig. 5 and 6, we may wonder whether one of the tested softwares performs particularly better than others. Taking into account the accuracy measure, Meshroom has the lowest error both in C2C and C2M in three out of four tests performed. However it is one of the slowest at reconstructing the objects. The number of reconstructed points is always smaller than the number of ground truth vertices (60,000 for the statue and 295,000 for the vase), but not the smallest within the tested softwares. Visually, the reconstructed models don't present particular issues. Colmap has a good accuracy almost in every test, except when using statue-ms scene. Its computational time is always under 10 minutes, which makes it one of the fastest. The number of reconstructed points is almost always the highest, twice as much as the ground truth in the statue case. Visually it is overall good, but it is compromised in the

TABLE V. Testing comparison between softwares in different object-scene conditions. The following metric is considered: absolute cloud-to-mesh distances (C2M) and absolute cloud-to-cloud distances (C2C) between the reconstructed point cloud and the provided ground truth. Mean distance values are reported in meters, with standard deviation in round brackets. Overall reconstruction time (in minutes) and number of reconstructed points are also reported. The dashed lines correspond to software failure in reconstructing.

Object-scene	Software	C2M	C2C	Time	Number of Points
Statue-fs	3DF Zephyr	0.018 (0.016)	0.036 (0.017)	11.23'	7,952
Statue-fs	Colmap	0.017 (0.031)	0.035 (0.030)	7.84'	130,176
Statue-fs	Meshroom	0.011 (0.038)	0.032 (0.038)	38.73'	34,000
Statue-fs	Metashape	-	-	-	-
Statue-fs	Regard3D	0.015 (0.044)	0.034 (0.044)	4.27'	70,079
Statue-ms	3DF Zephyr	0.087 (0.174)	0.097 (0.171)	9.77'	10,403
Statue-ms	Colmap	0.113 (0.244)	0.122 (0.242)	6.38'	149,182
Statue-ms	Meshroom	0.112 (0.250)	0.122 (0.247)	33.38'	38,013
Statue-ms	Metashape	-	-	2.69'	-
Statue-ms	Regard3D	0.164 (0.346)	0.172 (0.343)	2.91'	67,832
Vase-fs	3DF Zephyr	0.020 (0.038)	0.021 (0.038)	6.08'	8,023
Vase-fs	Colmap	0.018 (0.040)	0.018 (0.040)	8.026'	141,936
Vase-fs	Meshroom	0.016 (0.031)	0.016 (0.031)	54.12'	51,356
Vase-fs	Metashape	0.052 (0.033)	0.052 (0.033)	4,14'	89,982
Vase-fs	Regard3D	0.042 (0.054)	0.042 (0.054)	3.9'	52,868
Vase-ms	3DF Zephyr	0.007 (0.006)	0.007 (0.006)	6.32'	6,076
Vase-ms	Colmap	0.072 (0.079)	0.072 (0.079)	1.86'	23,801
Vase-ms	Meshroom	0.003 (0.003)	0.003 (0.003)	42.15'	42,025
Vase-ms	Metashape	-	-	3.68'	-
Vase-ms	Regard3D	-	-	-	-

vase-ms case, as it presents holes on one side. 3DF Zephyr has almost the same accuracy and time properties as Colmap, but it reconstructs the smallest number of points with respect to the other softwares. It doesn't present any particular issue visually. Regard3D is the fastest and fairly accurate, and it reconstructs as many points as the ground truth in the statue case. It doesn't present visible issues. In the vase-ms case it failed in the triangulation phase, thus not producing a reconstructed model. Metashape failed most of the times with the standard parameters, not reconstructing the objects of interest, but rather a ring of irrelevant points. The only scene it was able to reconstruct was the vase-fs, but visually the shape of the vase doesn't correspond faithfully to the ground truth shape.

With this analysis being made, we believe that the best software is 3DF Zephyr, as its main drawback is the fewer number of reconstructed points. All the other softwares present more severe issues (e.g. visual corruption, long reconstruction time or high failure rate).

B. Object and scene comparison analysis

With the same software, we can evaluate the results for different object geometry or light condition. Considering the statue, we can understand from Table V that the errors are smaller in the fs condition with respect to the ms one, by a factor of ten. This might be due to the more challenging process in the matching phases when the source of illumination is different across the images. However, this doesn't occur when considering the vase. A possible reason for this may be the simpler geometry of the object.

To give a fair comparison between the statue and the vase, at same light conditions, we need to consider that the model of statue is bigger than the one of the vase by a factor of ten. Taking this into account, the percentage error of the statue in fs scene is always smaller than the one of the vase. While in the ms scene, the percentage errors are similar.

VII. Conclusion

The contribution of this study is twofold. First, we give a comprehensive review of the passive 3D reconstruction problem, with particular focus on RGB images based methods for small objects. All the main categories have been analyzed: SfS, SfM and Deep Learning. The SfS approach has been described by carefully distinguishing all its sub-families, depending on the VH definition and its computation. SfM and Multi-View Stereo workflows have been analyzed, highlighting the common main steps and algorithms used in the pipeline. Some state of the art approaches have been discussed and categorized based on the strategies used in the process. Deep Learning based 3D reconstruction has been briefly introduced. In particular a possible categorization of the networks has been discussed and some of the recent algorithms have been categorized.

The second contribution of this study is a systematic comparison of the performances of the most widely used and commercially-available 3D reconstruction software. We highlighted their pros and cons, considering accuracy, processing time, number of points reconstructed and visual shape. Two different objects and lighting scenes belonging to a synthetic dataset were considered. Among the tested software, we think that 3DF Zephyr represents a valid trade-off between the considered metrics. Its main drawback, when compared to the others, is the small number of reconstructed points, which makes it not suited for applications where meshes and textures are paramount.

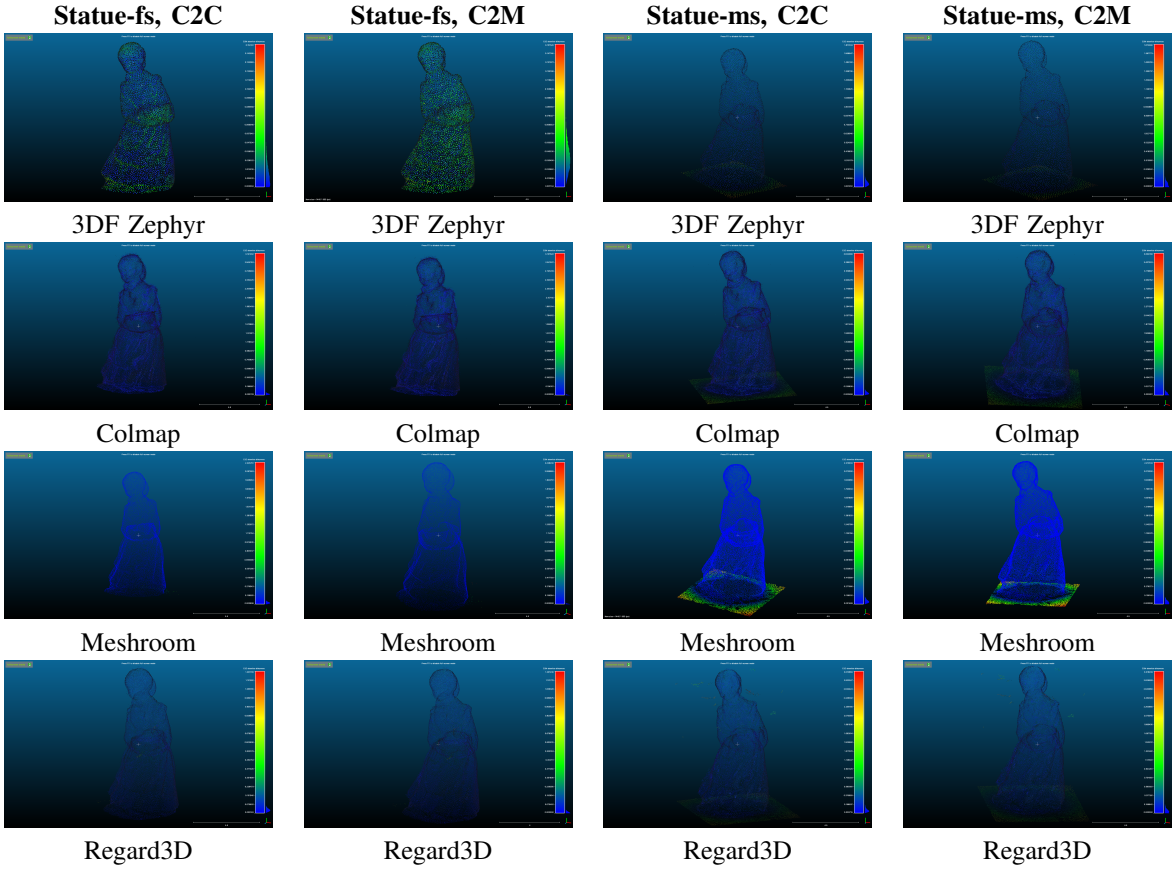

Fig. 5: A comparison of the residual reconstruction errors obtained by using different 3D reconstruction Softwares on the statue object. First two columns: fixed sunlight (fs) scene. Last two columns: moving sunlight (ms) scene. The first and third column show the cloud-to-cloud (C2C) distances, while the second and the fourth show the cloud-to-mesh (C2M) distances. The different rows display the output of the different softwares: from top to bottom, respectively, 3DF Zephyr, Colmap, Meshroom and Regard3D. Metashape failed at reconstructing the statues and thus it is not shown.

Possible future work could include a wider array of software in the comparison or, possibly, a richer amount of starting visual conditions. Without loss of generalization, however, the proposed work can be useful to the research community as a starting ground for 3D reconstruction studies.

REFERENCES

[1] M. Aharchi and M. A. Kbir, "A review on 3d reconstruction techniques from 2d images," in *The Proceedings of the Third International Conference on Smart City Applications*. Springer, 2019, pp. 510–522.

[2] J.-D. Durou, M. Falcone, Y. Queau, and S. Tozza, "A comprehensive introduction to photometric 3d-reconstruction," in *Advances in Photometric 3D-Reconstruction*. Springer, 2020, pp. 1–29. [Online]. Available: https://hal-normandie-univ.archives-ouvertes.fr/hal-02547952/document

[3] A. Prock and C. Dyer, "Towards real-time voxel coloring," in *Proceedings of the DARPA Image Understanding Workshop*, vol. 1. Citeseer, 1998, p. 2.

[4] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 369–374.

[5] G. Slabaugh, R. Schafer, and M. Hans, "Image-based photo hulls," in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*. IEEE, 2002, pp. 704–862.

[6] Cheung, Kanade, Bouguet, and Holler, "A real time system for robust 3d voxel reconstruction of human motions," vol. 2, pp. 714–720, 2000.

[7] M. Piccardi, "Background subtraction techniques: a review," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 4. IEEE, 2004, pp. 3099–3104.

[8] A. Laurentini, "The visual hull concept for silhouette-based image understanding," in *IEEE Transactions on Pattern Analysis and Machine Intelligence 16(2)*. IEEE, 1994, pp. 150–162. [Online]. Available: https://ieeexplore.ieee.org/document/273735

[9] Franco and Boyer, "Efficient polyhedral modeling from silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 414–427, 2009.

[10] G. Vogiatzis, C. Hernandez, and R. Cipolla, "Reconstruction in the round using photometric normals and silhouettes." in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1847–1854.

[11] A. Y. Mulayim, U. Yilmaz, and V. Atalay, "Silhouette-based 3-d model reconstruction from multiple images," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 4, pp. 582–591, 2003.

[12] G. K. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–375.

[13] G. Haro Ortega, "Shape from silhouette consensus and photo-consistency," in *2014 IEEE International Conference on Image Process-*

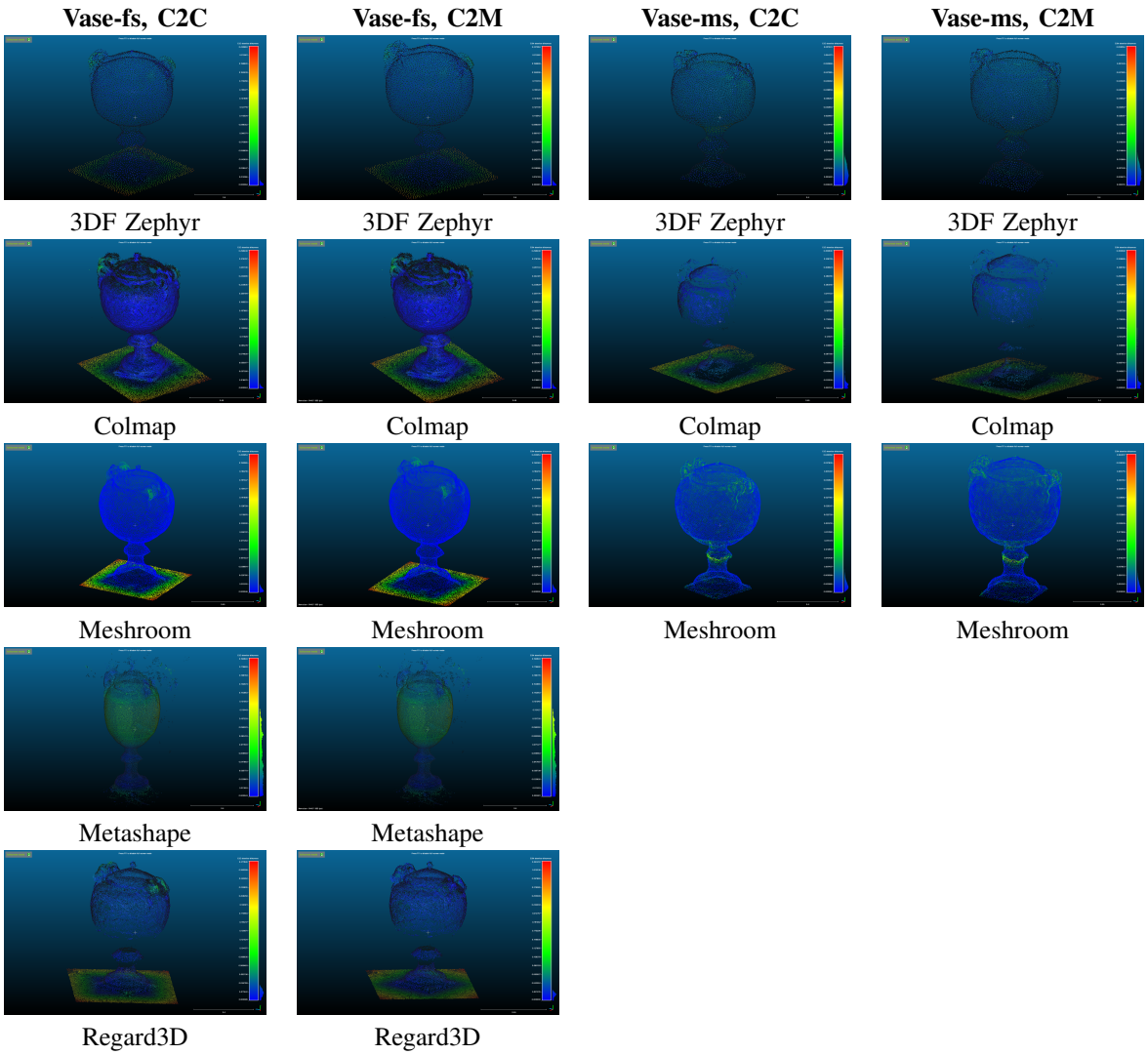

Fig. 6: A comparison of the residual reconstruction errors obtained by using different 3D reconstruction Softwares on the vase object. First two columns: fs scene. Last two columns: ms scene. The first and third column show the cloud-to-cloud (C2C) distances, while the second and the fourth show the cloud-to-mesh (C2M) distances. The different rows display the output of the different softwares: from top to bottom, respectively, 3DF Zephyr, Colmap, Meshroom, Metashape and Regard3D. Metashape and Regard3D failed at reconstructing the vase in ms condition.

ing (ICIP); 2014 Oct 27-30; Paris, France.[New York]: IEEE; 2014. p. 4837-41. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[14] C. H. Esteban and F. Schmitt, "Multi-stereo 3d object reconstruction," in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*. IEEE, 2002, pp. 159–166.

[15] S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini, and P. Perona, "3d reconstruction by shadow carving: Theory and practical evaluation," *International journal of computer vision*, vol. 71, no. 3, pp. 305–336, 2007.

[16] B. G. Baumgart, "Geometric modeling for computer vision," STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, Tech. Rep., 1974.

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

[18] Y. Yemez and F. Schmitt, "3d reconstruction of real objects with high resolution shape and texture," *Image and Vision computing*, vol. 22, no. 13, pp. 1137–1153, 2004.

[19] A. Erol, G. Bebis, R. D. Boyle, M. Nicolescu *et al.*, "Visual hull construction using adaptive sampling." in *WACV/MOTION*, 2005, pp. 234–241.

[20] D. Xia, D. Li, Q. Li, and S. Xu, "A novel approach for computing exact visual hull from silhouettes," *Optik*, vol. 122, no. 24, pp. 2220–2226, 2011.

[21] X. Liu, H. Yao, G. Yao, and W. Gao, "A novel volumetric shape from silhouette algorithm based on a centripetal pentahedron model," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 9–9.

[22] X. Liu, M. L. Gavrilova, and J. Rokne, "Incorporating object-centered sampling and delaunay tetrahedrization for visual hull reconstruction," *The Visual Computer*, vol. 25, no. 5, pp. 381–389, 2009.

[23] G. Haro and M. Pardas, "Shape from incomplete silhouettes based on the reprojection error," *Image and Vision Computing*, vol. 28, no. 9, pp. 1354–1368, 2010.

[24] G. Haro, "Shape from silhouette consensus," *Pattern Recognition*, vol. 45, no. 9, pp. 3231–3244, 2012.

[25] A. Tabb, "Shape from silhouette probability maps: reconstruction of thin objects in the presence of silhouette extraction and calibration error," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 161–168, 2013.

[26] L. Díaz-Más, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Shape from silhouette using dempster–shafer theory," *Pattern Recognition*, vol. 43, no. 6, pp. 2119–2131, 2010.

[27] J. R. C. Jose-Luis Landabaso, Montse Pardàs, "Shape from inconsistent silhouette," *Computer Vision and Image Understanding*, vol. 112, no. 2, pp. 210–224, 2008.

[28] E. Boyer and J.-S. Franco, "A hybrid approach for computing visual hulls of complex objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE Computer Society Press, 2003, pp. 695–701.

[29] M. Tarini, M. Callieri, C. Montani, C. Rocchini, K. Olsson, and T. Persson, "Marching intersections: An efficient approach to shape-from-silhouette." in *VMV*, 2002, pp. 283–290.

[30] S. Lazebnik, Y. Furukawa, and J. Ponce, "Projective visual hulls," *International Journal of Computer Vision*, vol. 74, no. 2, pp. 137–165, 2007.

[31] C. Liang and K.-Y. K. Wong, "3d reconstruction using silhouettes from unordered viewpoints," *Image and Vision Computing*, vol. 28, no. 4, pp. 579–589, 2010.

[32] H.-Y. Lin and J.-R. Wu, "3d reconstruction by combining shape from silhouette with stereo," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[33] W. Niem and J. Wingbermuhle, "Automatic reconstruction of 3d objects using a mobile monoscopic camera," in *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No. 97TB100134)*. IEEE, 1997, pp. 173–180.

[34] I.-S. Kweon, Y. Hwang, and J.-s. Kim, "Silhouette extraction for visual hull reconstruction," in *IAPR workshop on Machine Vision Applications (MVA)*, 2005, pp. 39–42.

[35] K. Forbes, F. Nicolls, G. De Jager, and A. Voigt, "Shape-from-silhouette with two mirrors and an uncalibrated camera," in *European Conference on Computer Vision*. Springer, 2006, pp. 165–178.

[36] Y. Xiang, S. Nakamura, H. Tamari, S. Takano, and Y. Okada, "3d model generation of cattle by shape-from-silhouette method for ict agriculture," in *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*. IEEE, 2016, pp. 611–616.

[37] D. Xia, F. Yang, and Q. Li, "Fast 3d modeling from images," *Optik*, vol. 124, no. 20, pp. 4621–4626, 2013.

[38] Z. Ma and S. Liu, "A review of 3d reconstruction techniques in civil engineering and their applications," *Advanced Engineering Informatics*, vol. 37, pp. 163–174, 2018.

[39] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O Connor, and J. Rosette, "Structure from motion photogrammetry in forestry: A review," *Current Forestry Reports*, vol. 5, no. 3, pp. 155–168, 2019.

[40] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[42] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[43] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[44] M.-D. Yang, C.-F. Chao, K.-S. Huang, L.-Y. Lu, and Y.-P. Chen, "Image-based 3d scene reconstruction and exploration in augmented reality," *Automation in Construction*, vol. 33, pp. 48–60, 2013.

[45] Y. Ham and M. Golparvar-Fard, "Epar: Energy performance augmented reality models for identification of building energy performance deviations between actual measurements and simulation results," *Energy and Buildings*, vol. 63, pp. 15–28, 2013.

[46] A. Khaloo and D. Lattanzi, "Hierarchical dense structure-from-motion reconstructions for infrastructure condition assessment," *Journal of Computing in Civil Engineering*, vol. 31, no. 1, p. 04016047, 2017.

[47] P. Rodriguez-Gonzalvez, D. Gonzalez-Aguilera, G. Lopez-Jimenez, and I. Picon-Cabrera, "Image-based modeling of built environment from an unmanned aerial system," *Automation in construction*, vol. 48, pp. 44–52, 2014.

[48] C. Wu, "Towards linear-time incremental structure from motion," in *2013 International Conference on 3D Vision - 3DV 2013*, June 2013, pp. 127–134.

[49] B. Bhowmick, S. Patra, A. Chatterjee, V. Govindu, and S. Banerjee, "Divide and conquer: Efficient large-scale structure from motion using graph partitioning," vol. 9004, 11 2014.

[50] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[51] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[52] M. Golparvar-Fard, V. Balali, and J. de la Garza, "Segmentation and recognition of highway assets using image-based 3d point clouds and semantic texton forests," *Journal of Computing in Civil Engineering*, vol. 29, 01 2012.

[53] C. Sung and P. Y. Kim, "3d terrain reconstruction of construction sites using a stereo camera," *Automation in Construction*, vol. 64, pp. 65–77, 2016.

[54] J. Zhang, M. Boutin, and D. G. Aliaga, "Robust bundle adjustment for structure from motion," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 2185–2188.

[55] B. Bhadrakom and K. Chaiyasarn, "As-built 3d modeling based on structure from motion for deformation assessment of historical buildings," vol. 11, pp. 2378–2384, 01 2016.

[56] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[57] R. Shah, A. Deshpande, and P. J. Narayanan, "Multistage sfm: A coarse-to-fine approach for 3d reconstruction," 2016.

[58] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 3248–3255.

[59] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello, "Hierarchical structure-and-motion recovery from uncalibrated images," *Computer Vision and Image Understanding*, vol. 140, p. 127–143, Nov 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2015.05.011

[60] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[61] M. Vergauwen and L. Van Gool, "Web-based 3d reconstruction service," *Machine vision and applications*, vol. 17, no. 6, pp. 411–426, 2006.

[62] G. M. Jog, H. Fathi, and I. Brilakis, "Automated computation of the fundamental matrix for vision based construction site applications," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 725–735, 2011.

[63] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 519–528.

[64] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment," *International Journal of Computer Vision*, vol. 84, no. 3, pp. 257–268, 2009.

[65] M. Smith, J. Carrivick, and D. Quincey, "Structure from motion photogrammetry in physical geography," *Progress in Physical Geography*, vol. 40, no. 2, pp. 247–275, 2016.

[66] M. Favalli, A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, "Multiview 3d reconstruction in geosciences," *Computers & Geosciences*, vol. 44, pp. 168–176, 2012.

[67] L. Wang, R. Chen, and D. Kong, "An improved patch based multi-view stereo (pmvs) algorithm," in *3rd International Conference on Computer Science and Service System*. Atlantis Press, 2014, pp. 9–12.

[68] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, "Mve—an image-based reconstruction environment," *Computers & Graphics*, vol. 53, pp. 44–53, 2015.

[69] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 1434–1441.

[70] Q. Xu and W. Tao, "Multi-scale geometric consistency guided multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5483–5492.

[71] "Agisoft metashape." [Online]. Available: https://www.agisoft.com/downloads/installer/

[72] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International journal of computer vision*, vol. 80, no. 2, pp. 189–210, 2008.

[73] "Pix4dmapper." [Online]. Available: http://pix4d.com/

[74] C. Wu, "Towards linear-time incremental structure from motion," in *2013 International Conference on 3D Vision-3DV 2013*. IEEE, 2013, pp. 127–134.

[75] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[76] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe, "Multilevel streaming for out-of-core surface reconstruction," in *Symposium on geometry processing*. Citeseer, 2007, pp. 69–78.

[77] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun, "Voronoi-based variational reconstruction of unoriented point sets," in *Symposium on Geometry processing*, vol. 7, 2007, pp. 39–48.

[78] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.

[79] X. Han, H. Laga, and M. Bennamoun, "Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2954885

[80] W. Shen, Y. Jia, and Y. Wu, "3d shape reconstruction from images in the frequency domain," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4471–4479.

[81] B. Yang, S. Wang, A. Markhan, and N. Trigoni, "Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction," in *Int J comput Vis*, vol. 128, 2020, pp. 53–73.

[82] W. Wang, Q. Xu, D. Ceylan, and U. Neumann, "Disn: Deep implicit surface network for high-quality single-view 3d reconstruction." Cornell University, 2019.

[83] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[84] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *ICCV*, 2019.

[85] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *3D Vision (3DV), 2017 Fifth International Conference on 3D Vision*, 2017.

[86] J. J. Georgia Gkioxari, Jitendra Malik, "Mesh r-cnn," *ICCV 2019*, 2019.

[87] S. Liu, T. Li, W. Chen, and 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.

[88] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," *European Conference on Computer Vision (ECCV)*, 2018.

[89] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent mvsnet for high-resolution multi-view stereo depth inference," *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[90] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, "Deepsfm: Structure from motion via deep bundle adjustment," in *ECCV*, 2020.

[91] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3d scene reconstruction from posed images," in *ECCV*, 2020. [Online]. Available: https://arxiv.org/abs/2003.10432

[92] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. Frahm, R. Yang, and M. Nistér, D. Pollefeys, "Real-time visibility-based fusion of depth maps," in *IEEE International Conference on Computer Vision*, 2007.

[93] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[94] [Online]. Available: https://www.3dflow.net/3df-zephyr-free/

[95] [Online]. Available: https://www.agisoft.com

[96] [Online]. Available: https://colmap.github.io

[97] [Online]. Available: https://alicevision.org/#meshroom

[98] [Online]. Available: http://www.regard3d.org

[99] D. Marelli, S. Bianco, and G. Ciocca, "Ivl-synthsfm-v2: a synthetic dataset with exact ground truth for the evaluation of 3d reconstruction pipelines," *Data in brief*, vol. 29, p. 105041, 2020.