3D Reconstruction of Cultural Heritages: Challenges and Advances on Precise Mesh Integration

Jurandir Santos Junior\textsuperscript{a}, Alexandre Vrubel\textsuperscript{a}, Olga R. P. Bellon\textsuperscript{a,}\textsuperscript{*}, Luciano Silva\textsuperscript{a}

\textsuperscript{a}Universidade Federal do Parana, IMAGO Research Group, Curitiba, Brazil

Abstract

Several methods perform the integration of multiple range scans of an object aiming the generation of a reconstructed triangle mesh; however, achieving high fidelity digital reconstructions is still a challenge. That is mostly due to the existence of outliers in the acquired range data, and their harmful effects on the integration algorithms. In this work, we first discuss artifacts usually found on real range data captured with 3D scanners based on laser triangulation. Following that there is the assessment of two widely used volumetric integration techniques (VRIP and Consensus Surface) and our suggested improvements. We also present a novel, hybrid approach that combines strengths from both VRIP and Consensus Surface, named IMAGO Volumetric Integration Algorithm (IVIA). Our novel algorithm adds new ideas while improving the detection and elimination of artifacts. Further, IVIA works in close cooperation with the subsequent hole filling process, which greatly improves the overall quality of the generated 3D models. Our technique leads to better results when assessed in different situations, when compared to VRIP, Consensus Surface, and also to a well known state-of-the-art surface-based method, Poisson Surface Reconstruction.

Keywords: 3D scanning, range data, shape reconstruction, volumetric integration

\textsuperscript{*}Corresponding author: Tel.: +55-041-33613683; fax: +55-041-33613205. E-mail address: olga@ufpr.br (O. R. P. Bellon).

Email addresses: jurandiro@ufpr.br (Jurandir Santos Junior), alexandrev@inf.ufpr.br (Alexandre Vrubel), olga@ufpr.br (Olga R. P. Bellon), luciano@ufpr.br (Luciano Silva)

Preprint submitted to Computer Vision and Image Understanding August 4, 2011
1. Introduction

Digital reconstruction of objects has several applications. Some, such as the preservation of natural and cultural assets, require high fidelity results at the cost of high-resolution range acquisition devices (e.g. laser triangulation 3D scanners [1]), along with a set of algorithms capable of dealing with noise and other artifacts present in range data.

Several steps compose a complete 3D reconstruction pipeline. First, data is acquired from different and sufficient viewpoints. Next, the data is aligned into a common reference frame in a process known as registration. After alignment, follows the mesh integration stage where the data are combined. Eventual holes due to incomplete data acquisition are usually filled after the integration step. Finally, a 3D model with its textures (i.e. diffuse color, specular and normal maps) is generated.

There are two surveys [2], [3] presenting an entire reconstruction pipeline. Other important works focusing on digital preservation of cultural heritage are presented in [1], [4], [5], [6]. They contribute by highlighting the difficulties to be overcome in complex objects scanning.

In this work, we focus on mesh integration within the reconstruction pipeline proposed in [6]. Several approaches perform this task; however, none is clearly superior. We implemented, tested and modified two state-of-the-art volumetric approaches, VRIP[4], [7] and Consensus Surface (CS) [5], [8]. In the attempt to solve their limitations, we developed a novel hybrid integration method, named IMAGO Volumetric Integration Algorithm (IVIA), which improves the fidelity of the generated 3D models. Our IVIA were briefly described firstly in [9]. But, in this work we present it in details and show its main results compared to other state-of-the-art approaches. Besides, we also suggest other improvements to Consensus Surface and explain the VRIP problem and our solution. Finally, as our IVIA was developed to deal with different types of artifacts, in this paper we analyze real case artifacts and use these difficult cases to assess the performance of our method.

We compare IVIA against two other volumetric integration algorithms to show how it overcomes the problems found in real cases and its capacity of generating more accurate results. In addition, we compare IVIA to the Poisson Surface Reconstruction (PSR) [10], which is considered the state-of-the-art surface-based method for 3D reconstruction, due to its good performance both in terms of memory usage and processing time.

To understand the challenge of integrating methods, first we review the
approaches for mesh integration in section 2. Then, in section 3 we present a qualitative analysis of the artifacts present on captured range data. In section 4, we begin by assessing two widely used volumetric integration methods (VRIP and Consensus Surfaces) and their variants, and suggest improvements; then we describe Poisson Surface Reconstruction. In section 5, in the attempt to solve the main limitations of VRIP and CS, we present in details a novel hybrid integration method, named IVIA. We also show how IVIA fully cooperates with the hole filling algorithm of Davis et al. [10]. We compare our results to VRIP, CS and PSR in section 6, followed by final remarks in section 7.

2. Related Works

There are several approaches for mesh integration; we will use the same categorization as in [2] to present them.

2.1. Delaunay-Based Methods

These methods use the Delaunay complex \( D(S) \) associated to a set of points \( S \) in \( \mathbb{R}^3 \). This complex imposes a connectivity structure to the points, and the methods extract a sub-complex of \( D(S) \) to represent the integrated surface. This class of algorithms works on point clouds, and Edelsbrunner [11] presents a review of them. We can highlight the techniques that use alpha-shapes [12]; power crusts [13]; cocones [14]; and eigencrusts [15].

The limitation of such algorithms is that they are sensitive to noise and outliers because they interpolate the data points. The solution for this requires a preprocessing step capable of “cleaning” the input data (e.g. [16]). Also, these algorithms are costly in performance, what usually limits the data set size that can be processed.

2.2. Surface-Based Methods

Some methods create or manipulate surfaces directly. In such cases, each range view defines a partial surface of the object, built from the triangulation of neighbor points in the captured grid of the range view.

The zippered meshes algorithm from Levoy and Turk [17] creates 3D models of the range views and erodes the regions at the intersection of the views (i.e. overlapping triangles). Then, the eroded regions are retriangulated and joined to build the final model. Soucy and Laurendeau [18] describe a
method where Venn diagrams are used to identify overlapping regions, which are later reparametrized and joined.

Bernardini et al. [19] proposed the ball-pivoting algorithm that works on point clouds, but without the need of Delaunay triangulation. The idea is to enlarge the surface from a seed triangle, generating additional faces through the “rotation” of a sphere around an edge from the perimeter of the region being created. Gopi et al. [20] calculates 2D Delaunay triangulations by projecting points over a plane, and later “elevating” these points in 3D to their correct position.

Some of these algorithms can fail in regions of high curvature, as shown in [7]. Topologically incorrect solutions can also occur due to outliers in the input views. Besides, some of these algorithms need a relatively constant sample rate in all views.

There is an additional problem. In general, these methods use fragments from the 3D models of each view. Without further post-processing, this generates noisy surfaces because each view is noisy. Therefore, by simply “attaching” pieces from different views does not prevent the noise from existing in them. That is not an adequate procedure, because each point on the surface has several samples, one for each view that observes it.

2.3. Parametric Surfaces

The idea here is to deform an initial approximation of the object through the use of external forces and internal reactions and constraints. We can imagine this as a “balloon” that inflates or deflates to assume the shape of the object. Other approaches use one or more analytically generated surfaces to represent the integrated model.

Terzopoulos et al. [21] use a deformable model where intrinsic forces induce a preference for symmetric shapes. Pentland and Sclaroff [22] adopted a method based on finite-elements and parametric surfaces by connecting “springs” between points on the surface with points of the object. The final model is obtained when the system stabilizes. Sharf et al. [23] use a deformable model in a coarse-to-fine approach, by tracking topological events to change the surface genus.

Carr et al. [24] use Radial Basis Functions to build the integrated surface from oriented point clouds. Ohtake et al. [25] present an algorithm where the final surface is produced by blending different parametric surfaces, generated and weighted locally. Fleishman et al. [26] improve this idea by using robust statistics.
Kazhdan et al. [27] calculate locally supported basis functions by solving sparse linear systems that represent a Poisson problem. A parallel and an out-of-core implementations of [27] are propose in [28], [29]. Other methods use level sets [30], [31]. They also employ robust statistics to eliminate noise and outliers. Most of these approaches use concepts from volumetric methods (e.g. implicit functions).

One limitation of parametric surfaces is the difficulty in representing sharp corners, as most methods assume a continuous differentiable function to represent the surface. Another related problem is finding a balance between over smoothing effect and non-elimination of noisy surfaces. They may also fill holes incorrectly.

2.4. Volumetric Methods

The basic idea of these methods is to create an implicit volumetric representation of the final model. So, each voxel has a value corresponding to the signed distance between the voxel and the integrated surface. The sign of the distance indicates whether the voxel is inside or outside the object. The object surface is defined by the isosurface at distance value 0. This isosurface is usually extracted with the MC (Marching Cubes) algorithm [32], [33]. This volumetric representation is also called a SDF (Signed Distance Field) and is broadly used in Computer Graphics. Unlike the parametric surfaces approach, the volumetric methods do not attempt to calculate the distance function analytically; it is defined solely through the interpolation of the samples of each voxel.

Several algorithms differ in calculating the distance function from the available data. Curless and Levoy proposed the VRIP (Volumetric Range Image Processing) [7] that calculates the distances according to the line-of-sight of the scanner and uses weights to assess the reliability of each measurement. The algorithm processes the views one by one and integrates the results into the SDF. To reduce time and space costs, only measurements taken near the surfaces are calculated and stored.

Hilton et al. [34] also combine data from various views, but use heuristics to know which measurements to consider in the calculation of the distance function. This is used to discard near views with opposite orientations, among other cases; however, these selection and discarding rules do not guarantee that all points belong to the same portion of the object surface, even in the ideal noiseless case. This assumption results in errors on the volumetric representation, as pointed in [8].
Wheeler et al. [8] presented an algorithm where outliers would be discarded through the calculation of a consensus surface. For a measurement to be considered in consensus there must be a minimum quorum of similar measurements. For each voxel, the consensus with distance closest to 0 is used, or in the case that no consensus was found, the voxel uses the non-consensus with largest quorum. To accelerate the algorithm, octrees are used, and the nodes of the octree become more subdivided when they are close to the 0 distance. This reduces both the memory use and processing time significantly. Improvements to the Wheeler’s algorithm were proposed by several papers from Sagawa et al. [35], [36], [37], [38].

Masuda [39] used a SDF to represent each view and performed a multi-view alignment and integration at the same time. The steps of alignment and integration are alternated until the result converges. As a result he can detect outliers and improve the overall quality of the process; however, his method is very expensive in terms of processing time and memory usage.

Rusinkiewicz et al. [40] used a real-time volumetric integration method on point clouds to allow an interactive preview of the object being scanned, to simplify the acquisition process. An offline post-process produces a high-quality final reconstruction using VRIP [7]. Rocchini et al. [41] presented the Marching Intersections algorithm to combine several range images. According to them, their algorithm has lower computational cost and provides better results than other volumetric methods; however, it is more sensitive to noisy input data.

Hornung and Kobbelt [42] use an unsigned distance field and a graph cut algorithm to extract the isosurface. The use of unsigned distances eliminates the need of normals for the input points. Yemez et al. [43] propose a volumetric fusion of range data acquired through optical triangulation and the technique of shape from silhouette. The advantage is to use the silhouette data to complete the reconstruction on occluded areas of the range data. One difficulty is to reliably extract the object silhouette without using a special illumination and background setup.

Volumetric methods have the advantage of using all available information (which helps to attenuate the noise of the input data) and ensuring the generation of manifold topologies [7]. Their main limitation is their performance, both in terms of memory usage and processing time. To achieve fidelity, the voxel size should be approximately equal to the scanner sample distance, which is usually small (around 0.33 mm). Small voxels generate large volumes; therefore great effort is necessary to make the processing of
these large data sets viable.

3. Range Data Analysis

To better understand the challenges of a high quality integration algorithm, we need to know the types of artifacts that appear in the input range data. In this section, we will analyze data captured with a laser triangulation 3D scanner (we used a Vivid 910 scanner from Konica Minolta). This type of scanner is considered one of the most precise currently available [44].

This analysis is important because several authors only add Gaussian noise to synthetic data to validate their techniques; however, real-case artifacts are more complex and harder to automatically detect and discard. One current avenue of research is dedicated to improve the quality of the acquired data. Nehab et al. [45] combine depth information from a triangulation scanner with normal information from photometric stereo. Park and Kak [46] proposed a technique to capture optically challenging objects through the usual laser triangulation technique with some modifications. Any improvement in the quality of acquired data is very welcome since it surely improves the fidelity of the 3D reconstruction.

In the following analysis, each range image has its point cloud triangulated into a mesh. Since the points are organized in a rectangular grid, it is easy to connect the vertices to their neighbors, creating then faces. This procedure has its own pitfalls, as explained in section 3.4.

3.1. Noise

The most common artifact on range data is noise on the object surface. Even for very smooth objects, the captured data is usually rough, as seen in Fig. 1.

The more powerful the laser, the rougher the captured surface tends to be. This is a problem because darker surfaces require greater laser power to be captured. Therefore, darker objects usually have noisier data than lighter ones. One way to attenuate this noise is through a weighted average from several samples of the same surface captured from different viewpoints.

3.2. False Data

Another common artifact is data returned by the scanner that do not exist in the object. They usually appear as small groups of triangles in regions that should be empty. They are caused by laser inter-reflections between
Figure 1: Noise in a captured view. The original object (a) has a smooth surface, but the captured data (b) is rough.

Figure 2: False data returned by the 3D scanner: (a) color image of a marble statue; (b) range image corresponding to (a), with false data generated by laser reflection between the statue and the table in the region of the elbow.

surfaces (as shown in Fig. 2), or when the laser is too oblique to the surface of the object, a common event in the silhouette of the object.

One way to eliminate this type of artifact is segmenting each view into connected components and discarding small regions; however, this procedure can lead to the elimination of valid data.

Another way is to observe that certain regions of the space are empty considering all views. Once the sensor position for each view is set, it is possible to define a volume between this point and the returned data that is known to be empty. By merging these volumes from all views, we are able to discard false data in empty regions. A similar idea was presented in [7], named space carving; however, they used such method to fill holes, and not to eliminate outliers, as proposed. This idea can be better understood through Fig. 3.
3.3. Deformed Surfaces

This is a more serious artifact, of difficult detection. Large surfaces, usually connected to valid data, are returned; however, they are completely wrong compared to the original object. We can see an example in Fig. 4.

This kind of artifact is usually caused by laser interreflections between the surfaces of the object (e.g. in corners) or the object and its supporting surface [47]. Whenever the object is more polished or reflexive, this artifact may appear more frequently (brighter surfaces are usually more reflexive than darker ones). Besides, the more powerful the laser used, the higher the chances of these reflections to occur. Unfortunately, this artifact is relatively common. Automatic detection and elimination of such artifacts is particularly difficult. Our IVIA approach successfully discards most of these artifacts, as shown in section 6.

3.4. False Silhouette Surfaces

Another problem is due to the triangulation of the point cloud returned by the scanner. The intricacy is to discern when adjacent points in the

Figure 3: Space carving being used to eliminate outliers. The empty regions from all views are merged, and any data inside these regions are discarded.

Figure 4: Deformed surfaces present on the returned data. The highlighted creases do not exist on the original object.
Figure 5: Elimination of false silhouette faces: (a) gross data returned by the scanner; (b) view surface without faces nearly parallel to the scanner line-of-sight.

returned data are not really neighbors but belong to different parts of the object. This occurs in internal and external silhouettes of the object. Our solution is to check the angle between the face normal and the line-of-sight from the scanner position to the center of the face. If this angle is above some threshold, the face should be discarded. A threshold of around 75° returned good results in all our experiments. We can observe this in Fig. 5.

This artifact may appear in another situation. The data returned by the scanner near silhouettes are usually unreliable. Sometimes, the borders near silhouettes are really deformed surfaces. A crude solution is to discard borders of the range view surface; however, this solution is far from being ideal: first, the artifacts are usually large (see Fig. 6), so they are not completely eliminated; second, lots of good data are discarded, impairing other stages of the reconstruction pipeline. Our suggestion is to treat these cases similarly to the deformed surface ones.

3.5. Impossible to be Captured Data

This problem is related to lack of data. As seen in Fig. 7, the interior of the fossil skull could not be captured through the fracture hole because it is an excessively deep recess for the scanner triangulation angle. Another source of the problem is the object material. An example is translucent materials as glass that cannot be captured because the laser propagates through them. Therefore, there are some regions on the surface of the object that cannot
Figure 6: Model of one range view with a large false silhouette surface highlighted. It behaves as a deformed surface, as described in section 3.3.

Figure 7: Incomplete data due to inaccessible regions to the scanner. (a) intensity image of a wolf fossil skull; (b) corresponding range image with occluded surfaces inside the skull. However, small unconnected regions are captured. That can possibly help the hole filling process.

be captured, resulting in holes in the reconstructed object.

For many applications, such holes are unacceptable and hole filling techniques complement the captured data [10]. One important consideration is that the more information we can extract from the captured data, the better the completed surface will be. While other integration techniques usually discard helpful data that could be used in the hole filling process, IVIA was designed to fully cooperate with the hole filling stage.

4. VRIP, CS and PSR Approaches

As presented in section 2, all integration approaches have limitations. Delaunay-based methods are costly, and require other techniques to eliminate bad data. Surface-based methods may have problems with complex topologies and are sensitive to noisy data. Most parametric surfaces algorithms
cannot handle sharp corners properly and surface fitting can be problematic due to the outliers. Finally, volumetric methods are costly in terms of time and space. In addition, Kazhdan et al. [27] compared several algorithms, and all of them were unable to ensure high fidelity reconstructions, especially at small-scale details.

We have chosen volumetric methods because they impose fewer restrictions to the reconstructed objects; offer an easy way to change the precision of the output (by varying the voxel size); can easily support the space carving technique, and can work in the presence of bad input data. Besides, they present better results compared to other techniques [27].

We developed our hybrid method by combining strengths from VRIP [7] and the CS [8] aiming to overcome their limitations while adding new ideas. So, in this section, we assess these volumetric approaches and suggest some improvements.

4.1. Volumetric Range Image Processing (VRIP)

Curless and Levoy [7] proposed a volumetric integration method named VRIP (Volumetric Range Image Processing) to calculate the signed distance from each voxel of the volume to the integrated surface. The essence of their method is the weighted sum of the SDF of each view, where the weight is used to represent the reliability of the measurements. This weighted sum generates an average surface for the integrated model. Some factors contribute to the weight: the angle between the surface normal and the scanner line-of-sight (measurement at grazing angles are less reliable); the distance to the view border and the absolute value of the calculated signed distance.

Another important observation is that the measurements are only assessed near the surfaces. Therefore, the distances are constrained to a range from $-D_{\text{min}}$ to $+D_{\text{max}}$. To achieve this, the weight approaches 0 when the distance nears these thresholds. We must also notice that the distances are calculated in the line-of-sight direction of each view.

The method processes the views one by one and integrates the results into a volumetric representation. Only measurements near to the surface are assessed and stored, resulting in the reduction of computational and memory cost. An interesting modification of the algorithm is called space carving [7]. Here, all voxels between the surface and the sensor are known to be empty. Empty voxels are stored with a distance $-D_{\text{min}}$ and weight 0. These empty voxels can be used later to help in the hole filling stage.
Advantages:

1. performs fast (O(n) regarding the number of views);
2. allows incremental addition of views (useful for successive refinements);
3. generates a smoother integrated surface, reducing noise due to the weighted average from all views;
4. uses the sensor position to define empty voxels, what is useful for hole filling algorithms;
5. does not require all views being loaded in memory simultaneously (views are processed one by one).

Disadvantages:

1. does not discard outliers. VRIP may reduce their weight, but this is difficult due to the fact that each view is processed individually;
2. integrates metrics from different viewpoints, what causes the SDF to be non-uniform. This may generate biased surfaces, and usually interferes with the hole filling stage;
3. causes artifacts in corners and thin surfaces. This is a critical problem, because it denies the generation of high fidelity reconstructions.

The third limitation of VRIP happens because distances behind the surfaces are not directly observed and in corners and thin surfaces they interfere with the correct measurements. In Fig. 8 we see the base of a statue that has a simple corner, but the reconstructed model introduced a circular crease (like a ring) near the bottom.

To understand how the crease is created, we need to examine slices of the volumetric representation. In Fig. 9c we can see a white corner (positive values) invading a dark gray region (negative values) of another view. VRIP combines these values and creates the creases, as indicated by the arrow in Fig. 9d. The same problem would happen on thin surfaces where one side would interfere with the other, increasing the thickness. As the algorithm requires the combination of positive and negative values (to find the average surface), it is difficult to eliminate these artifacts. They can be reduced by decreasing the values $-D_{\text{min}}$ and $+D_{\text{max}}$, but then integration fails when slightly misaligned views exist. Therefore, this is not a definitive solution for the problem.
Figure 8: Creases generated by the theoretical flaw of VRIP: (a) image of the object; (b) model reconstructed with VRIP, presenting a circular crease around the entire base.

Another problem is related to the effects caused by outliers, as presented in Fig. 10, where we did not fill the holes not to mask the artifacts of the integration.

4.1.1. Our Proposed Modification on VRIP

Fig. 11 illustrates the new weight attenuation curve that we propose according to the value of the signed distance. Curless and Levoy [7] suggested a curve with linear reduction from half of the range, but we adopted a non-symmetric curve, giving more weight to measurements of outside voxels. This curve aims to reduce both the influence of outliers and the problem near corners and thin surfaces. Our experiments show a reduction of artifacts in both cases; however, the position of the integrated surfaces changed slightly. So, we replaced one artifact for a less visible one; however it is not yet a definitive solution. This idea will be useful to our IVIA approach later.

4.2. Consensus Surface (CS)

The biggest limitation of VRIP is that it does not comprise any specific process to discard outliers. As we have presented, some “tricks” can be used to reduce their weight; however, even with lower weights, they still affect the final result. In our quest for a high fidelity algorithm, this behavior is not suitable. So we turned our efforts to the CS (Consensus Surfaces) algorithm of Wheeler et al. [8]. This algorithm also uses the SDF volumetric representation, extracting the resulting mesh with MC [32], [33]. It uses octrees to reduce computational and memory costs. In its original formulation all voxels near the surface are present; therefore, MC can be used without modifications.
Figure 9: Slices from the volume of the object in Fig. 8 (a) slice from a view; (b) corresponding slice from another view taken from another viewpoint; (c) slice from (a) layered over the slice from (b) to show their differences; (d) resulting slice of VRIP with all views merged. The isosurface is highlighted in red. Blue corresponds to the distance $-D_{\min}$ (empty voxels), brown to the distance $+D_{\max}$ (unobserved voxels), and the gray levels the distances from $-D_{\min}$ to $+D_{\max}$ (from black to white).

Figure 10: Effect of outliers on VRIP. The highlighted regions show (from left to right): 1) several creases generated by outliers from view borders; 2) wrong faces that are “floating” in the air; 3) completely irregular borders.
Figure 11: Distance weight curves for VRIP. The original curve [7] is shown with the dashed line, and our new curve with the solid one. Our new curve reduces both creases and the effect of outliers on the integrated surface. This factor ranges from 0.0 to 1.0 (according to the signed distance), and is multiplied by the other weight factors.

Sagawa et al. [35], [37] changed this approach by making unnecessary the division of the octree into its maximum level near the surface. This action requires an adaptation of the MC algorithm. The goal is to eliminate the outliers by taking into account only measurements consistent with each other. These consensual distances are averaged to create the signed distance to the surface from the current voxel. Another difference from VRIP is that here the distances are Euclidian, not over the scanner line-of-sight.

The algorithm is easy to understand. For each voxel of interest \( v \), we try to find an integrated candidate point with a consensus from several views. If this consensual point is not found, we use the non-consensual candidate with the largest weight \( w \). The candidates are integrated using the same weighted sum of VRIP, where the weight \( w' \) of each measurement is based on the angle of the surface normal \( n' \) and the scanner line-of-sight direction.

The idea is to generate a candidate point for each view by classifying it either into consensual or not-consensual. Each candidate starts at the nearest point to the voxel \( v \) in each view and is integrated with similar points on all other views. Notice that this step has a quadratic time complexity on the number of views, because each candidate must find its closest point on all other views and check if the point found is similar to the candidate (i.e. has similar surface normal and is sufficiently close). After all candidates are produced, the closest consensual candidate (if possible) is selected and called \( x \). Finally, the signed distance between \( v \) and \( x \) is calculated. To check if a candidate has consensus, the consensus threshold \( \theta \) is compared with the accumulated weight of the candidate.

In consensual regions from several views, CS eliminates outliers and provides a relatively smooth integrated surface. However, it performs slowly, their magnitude and sign of the distance may be incorrectly calculated around
the borders of views and when it automatically tries to fill holes, the results may present inconsistencies. We compare CS and VRIP algorithms in Fig. 12, and analyze below the limitations of CS.

Each object view is incomplete by definition; therefore, there will always be borders on the view models. The problem is that signed distances are ill defined to borders, because borders represent lack of information. So, when the nearest point to the voxel happens to be on a border, the magnitude of the calculated distance is greater than it should be, because the correct distance should have been calculated on the extension of the surface not captured in the view. Besides, we have neighbor voxels with large magnitudes and different signs. This will cause a surface to be generated on an arbitrary position by MC, which takes the magnitude into account when generating the triangle mesh. This generated surface is how the algorithm tries to automatically fill holes. During candidate integration, while these incorrect measurements are below the distance and angle threshold, they (incorrectly) affect the candidate position. When they exceed either of the thresholds, their effect suddenly stops. This can lead to roughness in the integrated surface.

Another common problem, mainly due to these border distances, is an incorrect estimation of the distance sign, as shown in [36], [38]. These incorrect signs lead to false surfaces being created near holes. To eliminate these problems, we believe we should always ignore distances to borders, as done in other works such as in [39].

We discovered in our experiments with CS that the consensus criterion does not always work as expected. If there is a far consensus, it has priority over a near non-consensus that is better to describe the correct surface of the object. The problem here is the choice of the consensus weight $\theta$: if it is low enough to turn the nonconsensus into consensus, outliers may not be eliminated and will affect the integration; if $\theta$ is too high, outliers are discarded but the problem described appears more frequently, usually causing holes in the reconstructed model.

Sagawa et al. [34], [36], [37] pointed that most of the visible artifacts on CS result from incorrect signs being calculated for the distance values. Also, in their solution, they ignore the magnitude of the distances (only in [37] they proposed a “patching” of the magnitudes to improve the quality of the hole-filled surfaces), focusing on “flipping” signs to improve the result. Besides, they assumed that the consensus criterion successfully removes all outliers. However, we noticed in our experiments that their magnitudes can
still be wrong, even if the SDF signs are repaired then, we cannot guarantee that the reconstruction will have high fidelity.

4.2.1. Our Proposed Modification on CS

We propose modifications on original CS trying to overcome its main limitations. We can both speed up the integration and solve the far consensus problem in a single action if we find the voxels near each view (we choose to use equal sized voxels to achieve the highest precision). We can both speed up the integration and solve the far consensus problem in a single action if we find the voxels near each view. To do that, we loop on each vertex of the view marking the voxels near them. Near, in this case, is defined as $+D_{max}$ of VRIP. Then, when generating the candidates for a voxel $v$, only views that marked $v$ are considered; consequently, far consensus will be automatically ignored, and the number of views processed for each voxel will be greatly reduced. Another benefit is that the calculation is made only for voxels near the object surface. Such action was achieved in the original CS through the use of octrees.

We choose to discard all of distance measurements when the nearest point in the view is on a mesh border, because CS calculates them incorrectly. We also multiply the CS weight by a border distance weight. It was suggested in [17] and used in VRIP. This helps to attenuate outliers and allows a
Figure 13: Modification on CS: (a) original algorithm; (b) our modified version. Hole-filling was disabled.

smoother blending of views. Another modification is related to producing the consensus candidates of each voxel $v$. Instead of executing searches for the nearest points on all views for each candidate, we only compare the candidates themselves, in order to accumulate consensus. So, we save a lot of kd-tree queries, and usually prevent the reload of the views into RAM when all views do not fit in memory simultaneously. This gives a great performance boost to the algorithm. Sagawa et al. [37] uses a similar approach.

To remove any eventual incorrect remaining data after the previous stages, we perform an additional data validation after the integration stage. This process discards voxels whose distances are not compatible with their neighbors, and it will be explained in more detail in section 5.

We can compare the result of the original CS with our modified version in Fig. 13, where we can see that almost all artifacts resulting of original CS integration were eliminated in our modified version. Even after our suggested improvements, CS still bears some original limitations, such as: the necessity to have all views loaded in memory simultaneously; the difficult in choosing thresholds and; outliers may appear on the integrated results as well. Aiming the generation of more accurate results, we developed our hybrid method, IVIA, by combining the strengths of both the VRIP and our modified CS. We describes IVIA in the following section.
5. The New IVIA Approach

In this section, we present our novel hybrid approach named IMAGO Volumetric Integration Algorithm (IVIA). IVIA was developed to automatically detect and discard several types of artifacts that usually appear on range data. Our approach performs the integration in two passes. First, a volumetric representation of the integrated model is created by using VRIP with our suggested modifications (see section 4.1); in its second pass, IVIA detects and discards outliers present in the representation created, and generates the final volumetric integration.

The pseudo-code of the first and second passes is presented in Figs. 14 and 15. The goal of the first pass is to build an approximated volumetric representation that will be stored on \textit{volCopy} through the modified VRIP presented in section 4.1. Besides, a binary volume \textit{empty} is created to represent the space carving. For each voxel there is a corresponding bit in \textit{empty}; if the bit is set, the corresponding voxel is considered empty (\textit{i.e.} outside the object).

Lines 1-17 of Fig. 14 correspond to the modified VRIP, plus the space carving. As the distance calculated in line 7 is over the line-of-sight of the scanner, any negative values are considered “empty” and the corresponding bit of \textit{empty} is set. It is important to notice that we use our modified distance weight curve in line 14, to help eliminate surface outliers. A threshold $+D_{\text{max}}$ should be defined: in our experiments, we use $+D_{\text{max}}$ as 3 to 5 times \textit{voxelSize}, with good results. After all views are processed, we apply a 3D mathematical morphology erosion operation on the binary volume \textit{empty}. This is necessary for two reasons: first, to prevent any incorrect measurement “dig holes” in the object. Though unusual, we already noticed this type of error on some range images. The other reason is that \textit{empty} is obtained through the union of empty spaces of all views. Therefore, it tends to represent the lowest measurement for each surface point, and not an average measurement. When reducing the empty space by a distance $+D_{\text{max}}$ we set aside a space near the surface to integrate the measurements from several views, and at the same time we keep an empty space representation to eliminate outliers far from the surface. Several artifacts presented in section 3 are eliminated by using \textit{empty} (lines 19-23).

Finishing the first pass, in line 24 we smoothen the values of the volumetric representation, using a 3 x 3 x 3 filter with larger weights to central voxels on the mask (\textit{i.e.} 2D smoothing filter). This smoothing completes individual
Figure 14: First pass of our proposed IVIA algorithm.

voxels with plausible values and attenuate noise and surface outliers. This attenuation is important because this volumetric representation will be used to estimate the integrated object normals in the second pass. In lines 25-32, the volumetric representation created is saved on \textit{volCopy} and reinitialized for the second pass.

The second pass of the algorithm does the definitive integration, discarding outliers. This pass possesses elements from CS, like the Euclidian distance calculation. We used them to prevent the biased surfaces generated when VRIP distances are used. Besides, Euclidian distances improve the hole filling algorithm we used [10]. Only voxels near the view surfaces and not set
Require: \(|-D_{\text{min}}| = |D_{\text{max}}|\)
Require: binary volume empty
Require: volumetric representation volCopy

1: for each view \(i\) do
2: Find the voxels near the view \(i\)
3: for each near voxel \(v\) do
4: if corresponding bit to \(v\) in empty is set then
5: continue to the next voxel \(v\)
6: end if
7: \(n\) ← estimated normal for voxel \(v\) from volCopy
8: if flagDiscardNoNormal is true and \(n\) then
9: continue to the next voxel \(v\)
10: end if
11: \(p'\) ← closest point to \(v\) in the surface of view \(i\)
12: \(n'\) ← surface normal of \(p'\)
13: \(w\) ← weight of \(p'\) in view \(i\) \(\text{as in VRIP}\)
14: \(d ← \|v - p'\| \\text{[absolute distance for voxel \(v\)}\]
15: if \(u' \cdot (v - p') > 0\) then
16: \(d ← d \\text{[voxel is outside the object]}\)
17: end if
18: if \((p'\) is in a border) or \((d > D_{\text{max}}) \text{ or (} \exists \text{ and } n' - n < \cos(\text{consensusAngle}))\) then
19: continue to the next voxel \(v\)
20: end if
21: \(w ← w * w_{\text{border}} * w_{\text{angle}} * w_{\text{distance}}\)
22: if \(d < -D_{\text{min}}\) then
23: \(d ← -D_{\text{min}}\)
24: end if
25: Update volumetric representation with \(d\) and \(w\)
26: end for \(\text{[each near voxel } v\}\)
27: end for \(\text{[each view } i]\)
28: Eliminate voxels with incompatible distances to neighbors

Figure 15: Second pass of our proposed IVIA algorithm.

in empty are evaluated. To find these voxels, we loop on each vertex of the view marking the voxels near them. This second pass has linear complexity on the number of views, instead of original CS, which is quadratic.

In line 7 of Fig. 15 an estimated normal \(n\) is calculated for the current voxel \(v\) from the volumetric representation volCopy generated in the first pass. This normal will be used to validate the view data, discarding incorrect measurements and outliers. The normal \(n\) is calculated from the \((x, y, z)\) gradients of volCopy. If the gradient magnitude is smaller than \(\frac{\text{voxelSize}}{3}\) or if the voxel does not have valid neighbors, \(n\) cannot be estimated. There are two possibilities in this case: if the boolean parameter flagDiscardNoNormal is true, the voxel is skipped; otherwise, this voxel will not be validated by the normal \(n\), which can lead to outliers being accepted on this voxel. The rare cases that require flagDiscardNoNormal to
be false are when there are very thin surfaces compared to the voxel size. In these cases, if $flagDiscardNoNormal$ is true, holes are usually created in the reconstructed surface. This parameter is taken into consideration in lines 8-10.

In lines 11-17 the nearest point $p'$, its normal $n'$, weight $w$ and signed distance $d$ are evaluated. These values are calculated as in CS by using a kd-tree of view $i$ to find the nearest vertex. The search for the nearest point is done on the faces incident on this vertex by calculating point to triangle distances. This process also tells us if the nearest point belongs to the border of view $i$.

The most important part of the algorithm is located in line 18. If $p'$ is located on a border, this measurement is discarded to avoid the problem discussed on section 4.2. Measurements larger than $+D_{max}$ are also discarded. Finally, if an estimated normal $n$ was calculated in line 7, the angle between $n$ and $n'$ is calculated. If this angle is larger than the threshold $consensusAngle$, the measurement is discarded. In our experiments, a value from $30^\circ$ to $45^\circ$ for the threshold $consensusAngle$ returned good results. This procedure solves the flaw of VRIP presented in Fig. 9. Besides, most of the deformed surfaces presented in section 3 are eliminated.

Another fundamental part of the algorithm is located in line 21. The basic reliability $w$ of the measurement (which depends on the angle between the scanner line-of-sight and the normal $n'$) is altered by other 3 factors. The first one is the border weight $w_{border}$, where measurements near the borders of the view have lower weights than interior measurements. This weight is used to discard $p'$ in line 18 if it is too small. The second factor is $w_{angle}$, due to the angle between $n$ and $n'$. This factor, ranging from 0.0 to 1.0, is calculated according to:

$$w_{angle} = \frac{(n.n') - \cos(consensusAngle)}{1.0 - \cos(consensusAngle)}$$

(1)

where, $n$ and $n'$ are normalized vectors. The value $(n.n)$ is used in line 18 to discard the measurement, therefore $1.0 \leq (n.n') \leq \cos(consensusAngle)$.

The third factor $w_{distance}$ varies according to the signed distance $d$, like the weight curves from VRIP, shown in Fig. 11. Our IVIA approach uses a new weight curve (see Fig. 16). This curve allows a smooth and unbiased integration of the distance values. Finally, we integrate the measurements of
all views through a weighted sum using the VRIP formula:

\[ D(v) = \frac{W(v)D(v) + w_i(v)d_i(v)}{W(v) + w_i(v)} \]

\[ W(v) = W(v) + w_i(v) \quad (2) \]

where \( D(v) \) is the accumulated distance for the voxel \( v \), \( W(v) \) is the accumulated weight for \( v \), \( d_i(v) \) and \( w_i(v) \) are respectively the distance and weight of \( v \) in view \( i \).

In line 28, after the integration is completed, we have a last error elimination step. Neighboring voxels should have similar distance measurements because of the use of Euclidian distances. Therefore, two neighbor voxels should have a maximum distance difference equivalent to the distance between the voxel centers. Using 26-neighborhood, the distances between voxel centers can be \( \text{voxelSize} \), \( \sqrt{2} \cdot \text{voxelSize} \) or \( \sqrt{3} \cdot \text{voxelSize} \). However, we cannot be so restrictive, because the integration slightly violates this condition, due to the several weights used. Nevertheless, neighboring voxels with excessively different values should be discarded to avoid the generation of bad surfaces on the final model. Therefore, we use a threshold \( \text{compatibleFactor} \) (usually 1.5), which multiplied by the voxel center distance gives us the maximum allowed difference between neighboring voxels. An important detail is that this elimination cannot be done in a single pass, because a “wrong” voxel ends up spoiling its neighbors, since their difference would be too large. Because of that, we first gather all the “suspect” voxels, and sort them (using \text{bucketsort}) by the number of wrong neighbors (which can be 26 at maximum). Next, we eliminate the “most suspect” candidate, updating the suspect list to ignore neighborhood to voxels already eliminated. Therefore, the elimination is more selective, discarding only the really incompatible voxels. These voxels are tagged as unknown (\( i.e. \) distance value of \( +D_{\text{max}} \) and weight 0).

We developed IVIA to deal with noise and outliers in several ways. So, we eliminate outliers far from the surface by using the space carving technique. The first pass of IVIA also eliminates outliers near the surface and measurements out of consensus. Further, the second pass helps to decrease the influence of any outliers that had not been removed previously by assessing the reliability of the measurements through several parameters. Besides, we eliminate the remaining incompatible data after integration.
6. Experimental Results

In this section, we present the experimental results we performed aiming to assess the effectiveness of IVIA. Our main concern is to analyze its accuracy for applications such as digital preservation of cultural and natural assets, that demand high fidelity 3D reconstructions. The dataset used in the experiments is composed of art objects (from the Metropolitan Museum of Curitiba), fossils (from the Natural Science Museum of UFPR), insects (from the Biological Collections of UFPR), Baroque masterpieces (Aleijadinho’s sculptures) and personal objects.

A few IVIA’s results were presented in [9], as mentioned in section 1; however in this section we present our main results. Our goal was to assess five main points: outliers’ removal, comparisons to other volumetric algorithms, cooperation with the hole filling process, comparison to the state-of-the-art surface-based method PSR and IVIA’s performance and resolution. The results presented here show how our method deals with difficult situations when compared to the other methods.

6.1. Outliers’ Removal

In Fig. 17 one can see an example of the ability of IVIA to handle outliers. Fig. 17a illustrates a detailed region (the boy’s arm) of the reconstructed model of the sculpture shown in Fig. 2a. We can see the low quality (several artifacts) of the correspondent captured view in Fig. 17b. The artifacts happen because the statue material (marble) is not optically cooperative. Nevertheless, our hybrid algorithm eliminates or reduces most of the artifacts, then a more suitable model is generated. We show in Fig. 17c the influence of the \( w_{\text{angle}} \) factor on the view data.

6.2. Comparison to VRIP and CS

We applied the three volumetric algorithms in a difficult case (sculpture shown in Fig. 2), and in this section we show a comparison of the obtained
results using Figs. 18 as support. VRIP results (Figs. 18a and 18d) are good, but outliers still remain on the final model (red arrows) as well as the problems on edges and thing surfaces (see section 4.1). CS returned few outliers on the final result, as shown in Figs. 18b and 18e, even after applying our suggested modifications. Further, CS did not create a smooth surface. So, we see that even after using a postprocessing step to discard disconnected geometries, the results from both algorithms still have limitations.

Figs. 18c and 18f illustrate the model generated by using IVIA. We see that the model surface obtained by IVIA is smoother than the CS result. Also, IVIA results do not present the problems such as the ones seen on VRIP’s model. In addition, we also see that our hybrid algorithm eliminated/reduced incorrect data successfully. Besides, IVIA generates a more reliable model while we pay the price of slightly larger holes by eliminating unreliable areas.

6.3. Hole Filling

As mentioned IVIA was developed to cooperate with the hole filling process. We use the VD (Volumetric Diffusion) method of Davis et al. [10] with very good results, as shown in Fig. 20. Reliable results were generated on all cases, and the holes filled in a predictable way. Sagawa et al. [36], [38] compared their modification of CS with VD, showing very bad VD results. However, as shown here, VD performs well. The problem is the sensitivity of VD to outliers. As it propagates information to fill holes, bad data present after the integration severely impairs VD.

26
Figure 18: Comparison among 3 integration algorithms. Reconstructed marble statue and a detailed region: (a),(d) VRIP; (b),(e) our modified CS; (c),(f) IVIA. In (d), we can see several artifacts (indicated by arrows), but in general the surface is smooth. The reconstruction in (e) presents a slightly rougher surface, but the most obvious artifacts were eliminated. In (f), IVIA eliminated almost all incorrect data, kept the surface smooth, though it generated more holes than VRIP, but comparable with the ones from modified CS. Hole filling was disabled in all cases.
Besides a more effective outlier removal (see section 6.1), our use of Euclidean distances solves another problem mentioned on [10], namely the incorrect surface extension, as we show in Fig. 19. If used with VRIP, the hole borders are usually sampled at an angle due to the scanner line-of-sight distance measurement; this causes incorrect results after the volumetric diffusion. By combining Euclidean distances with the disposal of measurements to view borders in IVIA, one can overcome this limitation. Finally, the space carving we perform aids the hole filling as well, as explained in [10], producing more accurate results. Our IVIA has the benefit to use all the information available cooperating with Davis’ algorithm. However, PSR does not use all available information as well (e.g. line-of-sight information) what may cause incorrect connections of some regions (see Fig. 20c), a problem pointed out by the authors [27].

6.4. Comparison to PSR

We activate hole filling aiming to compare our results to PSR (authors’ original). Fig. 20c compares the IVIA’s reconstructed wolf fossil with holes filled (up) to PSR’s one (down). This is the same fossil illustrated in Fig. 7 with incomplete data in some regions. As our IVIA uses all information that can be deduced from range data it integrates the fossil skull and cooperate with VD algorithm during hole filling. In the highlighted regions in the same
Figure 20: Results of IVIA integration: (a) details of Joel’s vestment; (b) marble statue from Fig. 18 (up) and zoolito (down); (c) wolf fossil skull, result for IVIA (up) and PSR (down). Some regions difficult to be filled are highlighted.

Fig. (down), we see that PSR fills holes incorrectly as it does not use all available information.

We also compare IVIA to PSR with large models, such as Aleijadinho’s Prophets. These near real size sculptures were made in the beginning of the XIX Century, hand-carved in soapstone. They represent twelve of the sixteen prophets from the Holy Bible and are part of the Bom Jesus de Matosinhos Sanctuary, a UNESCO World Heritage Site [48]. The IVIA’s reconstructed model of Aleijadinho’s Prophet Joel is presented in Fig. 21b.

The PSR difficult to find a balance between over smoothing effect and non-elimination of noisy surfaces can be seen in Fig. 21c. This figure shows the parchment of Joel in details, where we can see that PSR reconstructed model (Fig. 21c down) is over smoothed compared to our IVIA (Fig. 21c up). We also noticed in the highlighted rectangle area in Fig. 21d (down) that PSR creates outliers that appear as vertices. It happens because, PSR does not ensure the generation of manifold topologies and when it creates manifold meshes it may occur that these outliers appear in the final model. IVIA’s result for Joel’s face is shown in Fig. 21d (up).

6.5. Performance and Resolution

Table 1 shows the temporal efficiency of IVIA on models with different sizes compared to VRIP and PSR. The computer used was a 1.86GHz Core 2 Duo PC, with 4 GByte of RAM. When reconstructing large models (as
Figure 21: Aleijadinho’s Prophet Joel: (a) model with high resolution texture; (b) complete reconstructed model; (c) comparision between IVIA and PSR reconstruction of Joel parchment; (d) comparision between IVIA and PSR reconstruction of Joel head.
<table>
<thead>
<tr>
<th>Model</th>
<th>Faces</th>
<th>Voxel</th>
<th>IVIA</th>
<th>VRIP</th>
<th>PSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoolito</td>
<td>317610</td>
<td>0.5</td>
<td>498</td>
<td>121</td>
<td>231</td>
</tr>
<tr>
<td>duck</td>
<td>206627</td>
<td>0.8</td>
<td>900</td>
<td>227</td>
<td>718</td>
</tr>
<tr>
<td>wolf</td>
<td>954938</td>
<td>1.0</td>
<td>1291</td>
<td>1220</td>
<td>1142</td>
</tr>
<tr>
<td>statue</td>
<td>1289654</td>
<td>1.0</td>
<td>3600</td>
<td>458</td>
<td>1832</td>
</tr>
<tr>
<td>Joel</td>
<td>3487652</td>
<td>2.0</td>
<td>11265</td>
<td>3050</td>
<td>3217</td>
</tr>
</tbody>
</table>

Table 1: The running time comparison. For each model are shown the number of faces, the voxel size (in mm) used for reconstruction with IVIA and the running time (in seconds) for IVIA, VRIP and PSR.

Prophet Joel) the reconstructing time of our method increases. In these cases, our IVIA algorithm, although being slower than VRIP and PSR, can still be further optimized using 3D scan conversion on the space carving stage, the stage when the algorithm spends most of its execution time. The modified CS efficiency is low, when taking into account both running time and the quality of the results. The results for original CS and our modified CS was not feasible, their running times for the marble statue model were 400980sec (original CS) and 70740sec (our modified CS).

As we aim to reconstruct high fidelity models with high resolution, the IVIA’s steps to remove outliers and deal with the artifacts addressed in section 3 is essencially important. Although the running time is increased, we can reach good results. Fig. 20a shows a detailed region of Aleijadinho’s Prophet Joel vestment, reconstructed at a high resolution. Another results can be seen in Fig. 20 and comparisons to PSR presented in Fig. 21. We must notice that the cases shown pose challenges: thin surfaces, optically uncooperative materials, sharp corners and complex topologies with lots of occlusions. The IVIA algorithm was able to overcome all of them. We also illustrate in Fig. 21d the Joel model with high resolution texture, obtained by using the texture generation method from [49].

7. Final Remarks

In this work, we analyzed the complex types of artifacts in real input range data. Also we discussed the limitations that widely used volumetric integration methods may have. Trying to overcome these limitations to achieve high fidelity digital reconstructions, initially we proposed enhancements to both
VRIP and CS, then we presented our novel IVIA algorithm. IVIA achieved precise reconstruction results compared to other state-of-the-art algorithms.

Our hybrid approach, IVIA, may still be improved. As we use VRIP in the first pass, outliers that survive this pass can be accepted later. Trying to solve this problem we reduce the outliers’ influence by mixing them with good measurements. However, in very noisy regions, this methodology may still fail. Despite of that, IVIA yields good experimental results. One possibility to solve this problem is to use more passes to improve outlier detection and removal between passes. Another alternative could be the use of a combination of parametric surfaces (as in Ohtake et. al [25]) with a volumetric representation and our outlier detection and removal techniques. Finally, to the best of our knowledge, there is no suitable metric to quantitatively compare 3D reconstruction results, so this could be an interesting avenue of research.

8. Acknowledgments

The authors would like to thanks to CNPq, CAPES, UNESCO and IPHAN for supporting this research.

References


