3DSAINT Representation for 3D Point Clouds

Chandra Kambhamettu
Video/Image Modeling and Synthesis (VIMS) Lab.
Computer and Information Sciences, University of Delaware
Newark, Delaware 19716
chandrak@udel.edu

Abstract

This paper introduces a Sphere-based representation to model a 3D scene and show its performance on various tasks, including Structure from Motion (SfM) and 3D scene classification. A significant target application of this work is Mixed Reality, where 3D data can be efficiently represented, and synthetic and real data can be mixed for an immersive experience. Over the past few decades, 3D big data has garnered increased attention in computer vision. Acquiring, representing, reconstructing, querying, classifying, and visualizing 3D models for Mixed Reality has become crucial for many applications, such as medicine, architecture, entertainment, and bioinformatics. With the ever-increasing amount of data that the 3D scanners produce, storing, processing, and transmitting the data becomes challenging. Techniques that exploit the shape information need to be developed to model, classify and visualize the data. Our work offers a novel multi-scale surface representation based on spheres, with the ultimate goal of helping scientists to see and work with 3D data in Mixed Reality more effectively and efficiently.

1. Introduction

In computer vision, [30], the representation is all about how the input, output, and intermediate information are constituted for the algorithms to calculate the desired results. The utility of 3D points in computer vision, computer graphics, mixed/virtual reality, medical imaging, and multimedia has recently increased multi-fold. Visualizing data in 3D enables observers to examine and interact with objects as they appear in reality. Therefore, an efficient and effective representation of the 3D world is imperative in mixed reality for storage, analysis, recognition, transmission, and visualization.

Real-time 3D capture using RGB-D cameras and other multi-view setups has become common. The growth in 3D sensing is due to various advantages that are well-known. The most obvious reason is that we live in a 3D world; for robots and other artificial intelligent agents to work together with naturally intelligent agents (such as humans and other biological organisms), 3D scene acquisition and understanding are needed. Furthermore, all our natural changes, such as the world's growth and dynamics, happen in 3D. Some examples that make it imperative to analyze in 3D include autonomous vehicle navigation, biological organ modeling, natural hazards and environmental analysis, and sports analysis. 3D sensors are now smaller, compact, and less bulky [11]. They can be hung on a Drone or a robot for environment sensing and easy navigation. We can notice how the 3D gaming industry provides a more impressive and immersive viewing experience using TVs, smartphones, and tablets with miniaturized sensors, by sensing and tracking faces, hands, legs, and body movements and providing a life-like mixed reality environment [16]. Due to the prevalence of Laser Scanning (LiDAR) and unmanned aerial vehicles, Mixed Reality has seen a lot of applications in Geosciences [12].

Computer vision methods are needed to model the 3D big data involved in Mixed Reality systems, which is further poised to increase exponentially. In Graphics, mesh shading is usually done using meshlets, representing a local mesh with a predefined upper limit of V vertices and P primitives. Alternatively, in vision-related tasks, to model a set of 3D points constituting a scene, the point cloud is converted to a primitive-based model by the use of 3D primitives (oriented 3D rectangles, i.e. cuboids) [41]. Such primitive-based 3D object representations have been utilized before for modeling with underpinnings from psychology [3]. Once obtained, these primitives can be used for storing, transmitting, recognizing, and retrieving. In this paper, we introduce a Sphere based representation to model a 3D object/scene. There are several advantages to using spheres as our modeling primitive. First, spheres do not have a boundary, unlike bounded planes. In Computer Graphics, Spheres have been utilized for modeling deformable surfaces [29]. As the authors indicate, spheres have the main advantage of having a

continuously varying tangent plane and well-defined curvature at every point.

There are several sources of 3D data; some 3D sensors include i) LIDAR, a 360° omnidirectional scanning device used widely for autonomous vehicle navigation. Depth is obtained by measuring the reflected pulses from the target object; ii) Stereo and multi-view cameras, which use point correspondences between different views to get the 3D or by projecting light patterns into the scene (Intel's DEPTHSENSE Camera series [15]); iii) Use of IR sensor to reconstruct the depth map of the scene (Microsoft's Kinect series). For example, the Kinect V2 uses Time of Flight information instead of the previous Light Coding to produce a 512 by 424 depth map and a visible camera to produce 1920 by 1080 full HD color image [20, 21]. Other avenues of 3D information is from the Structure From Motion (SfM) algorithms, as well as other deep learning based architectures [38], and NeRFs [19].

2. Background and Related Work

The entertainment, aviation, real estate, defense, healthcare, and education industries are some of the major stakeholders of MR technologies. Real estate companies like to show the visualization of 3D structures to prospective clients, even before breaking the ground. In healthcare, doctors can store the 3D data from surgeries and simulations; the sports industry can visualize the events and keep them for further viewing. All these exciting applications create a tremendous amount of data that must be securely stored for the long term. Currently, the file sizes are enormous, and most of these applications generate up to one terabyte of data per hour; for instance, a 3-hour game would be three terabytes.

Mixed Reality must have a compact and efficient 3D model to store the vast data, facilitating in-situ embedding of 3D models of scene geometry, playback of recordings, etc. Retrieving models from large databases also needs an efficient, compact representation of 3D shapes and robust matching [36]. 3D point sets consist of x,y,z coordinates; when no adjacency is known, it is unstructured, and when provided, it is in a mesh format where data is structured. Points can describe standalone objects or a conglomeration of objects comprising a scene. The number of points can run into millions, and given that each coordinate contains three floating-point numbers (x,y,z), storage itself will be costly. This issue is more so for dynamic scenes, where we have 3D data in multiple frames due to the temporal dimension. Transmission of this data is not feasible in real-time for practical applications. In remote surgery applications alone, the encoding of comprehensive 3D data is needed for compression, storage, transmission, and rendering. We now review 3D point processing research work and categorize it into representation, registration, and classification.

Several works have been presented in representation and used in shape retrieval. Authors have proposed a novel tool, called the Spherical Harmonic Representation, that transforms rotation-dependent shape descriptors into independent rotation ones, helping design practical shape retrieval algorithms [13]. Deep learning methods (Siamese) were used to perform 3D matching. In [34], authors perform sketch-based shape retrieval using such a deep network. In [33], authors use Sphere Region Proposal Network (SphereRPN) to detect objects by learning spheres instead of bounding boxes and showing the robustness towards localization error compared to bounding boxes.

In registration, there has been work on establishing reliable 3D shape correspondences between 3D scans, using a template 3D shape. Authors use *Shape Deformation Networks*, a comprehensive, all-in-one solution to template-driven shape matching [7]. A *Shape Deformation Network* learns to deform a template shape to align with an input observed shape. Given two input shapes, the authors align the template to both inputs and obtain the final map between the inputs by reading off the correspondences from the template. Spherical harmonic cross-correlation is another robust registration technique based on the normals of two-point clouds with significant overlapping regions. Since this technique has a high computational cost of computing spherical harmonics at each normal, the binning of normals has been proposed by earlier work [17]. Such binning improves computational efficiency since the spherical harmonics can be pre-computed and cached for each bin location. In [18], authors estimate rotation by traversing the space of rotations to obtain a maximum correlation between Extended Gaussian Images (EGI) of the two 3D datasets. This is efficiently computed using the spherical harmonics of the Extended Gaussian Image and the rotational Fourier transform.

In classification, 3D point processing has recently received much attention, thanks to PointNet [24]. In this work, the authors design a novel type of neural network that directly feeds on point clouds with permutation invariance. Their network consists of a unified architecture for applications composed of but not limited to, object classification, part segmentation, to scene semantic parsing. However, PointNet does not capture local structures in the metric space of points, so it is not generalizable to complex scenes. Thus the authors have introduced PointNet++, a hierarchical neural network that applies recursively on a nested partitioning of the input point [25]. Several 3D point cloud networks came after that, including the work on 3D object detection [23]. Several datasets are available for benchmark evaluation, such as the synthetic dataset, the ModelNet40 [35]. A recent survey in this series of works [8] presents a taxonomy of current methods.

Limited work has been done on the security aspects of

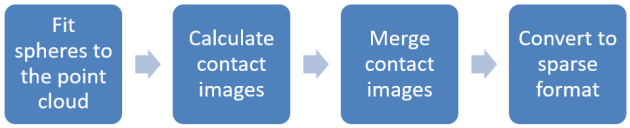

Figure 1. 3D Surface Aggregation in Topology Representation (SAINT)-based modeling pipeline.

3D Big data. A data breach is when an unauthorized user gains access to valuable and confidential data, either measured by scanners or generated by algorithms. For example, our work can be applied to results from work such as [32], who proposed a robust method for regressing discriminative 3D morphable face models (3DMM). The authors use a convolutional neural network (CNN) and regress 3DMM shape and texture parameters directly from input intensities. However, one of the significant disadvantages of storing 3D points is the inefficiency, and unsafe against Cyber threats as well. Our system can convert these output 3D faces, and other such works [10] into sphere-based representations to store and transfer efficiently and securely. For example, if data transmitted is stored as 3D points, it is prone for hackers to understand and 'attack' the data. Using our work, we can encode the data into spheres (radii and centers with our method), oblivious to the actual 3D data. Then, with the corresponding contact map, 3D points can be reconstructed from sphere centers and radii. Moreover, the critical spheres center data can be encrypted and transmitted/stored securely, whereas contact maps can be transmitted/stored 'publicly.' A meshlet, representing a variable number of vertices and primitives used in graphics pipelines, is not encrypted and is thus open to a data breach.

3. Our Approach

3.1. Overview

Our sphere-based modeling approach iteratively fits spheres to the input point cloud (Figure 1). The resulting shape model is multi-scale as the number of spheres utilized can be controlled, and details of the shape captured are multi-scale. We term this method as 3D SAINT, Surface Aggregation in Topology Representation. We later show that this modeling scheme successfully improves many of the 3D tasks , such as classification. Given a 3D point cloud as input, our method computes a series of spheres as its surface approximation. In [4], authors fill the inside or outside of the 3D model with an appropriate number of infilling spheres. A sphere in their work is defined by a voxel with the voxel coordinates as its center and the signed distance field (SDF) value as its radius. In our method, we use spheres to model the surface of the point cloud. Our shape modeling aims to maximize geometric accuracy and, simultaneously, is straightforward regarding the number of parameters. The main goal is to simplify the surface representation from points to a low number of parameters. As a result, we can reconstruct the shape and transmit quickly, store it efficiently, and classify it accurately using our representation.

A shape is usually defined as all that information, excluding its location, orientation, and scale [14]. A sound shape analysis system aims to estimate a finite subset of shapes (*sample*) from a population of shapes that best represents the shape. We propose a representation scheme for surfaces, mathematically defined as a smooth manifold with co-dimension one, embedded in a Euclidean space. The analysis of shape based on parameters corresponding to a base shape (sphere in our case) is also an essential tool to study the relationship between different shapes, such as normal vs. abnormal anatomy in a medical domain.

As shown in Figure 1, our algorithm receives input from a 3D point cloud, a user-specified number of spheres (upper limit, *ms*), as well as sampling resolution (*mr*) in the polar domain. Both these factors (*ms* and *mr*) can be utilized for coarse-to-fine scale representation of the surface in question. Sphere-based representation of shapes is one of the fundamental ways to capture 3D shapes. Mapping 3D data onto spheres makes this analysis attractive, as spatial relations can be quickly apprehended, especially when the viewpoint is rotated. In some sense, (rigid and non-rigid) motions to 3D data are equivalent to motions to the spheres. Our proposed shape representation can be obtained to compactly encode 3D shapes while enabling a descriptive model of the shape in terms of the number of spheres and their radii. We also become relevant to works such as [40], who build patch-based surface CNNs, which our work can provide through our spheres parametrization. Thus, we provide compression as well as parametrization, thanks to the representation by a sphere at each point. We, moreover, do not need a 3D point cloud in a mesh format. These are some of the many reasons our method improves over techniques based on mesh simplification strategies [6] or Project DRACO from google. Draco is a library for compressing and decompressing 3D geometric meshes and point clouds, geared towards improving the rendering, storage, and transmission of 3D graphics [1].

As shown in (Figure 2), we first fit the sphere to the surface data; based on the implicit error threshold, we decide what points to keep (belonging to that particular sphere) and then use the remaining points to fit the sphere again. This process is done iteratively until very few points remain (around .1% of total points.) Each fit generates a set of inliers and outliers. We record the position of inliers in the corresponding sphere's polar coordinates and call it *contact map or contact image*. Thus we will have a set of spheres' radii and corresponding *contact maps* representing a given surface.

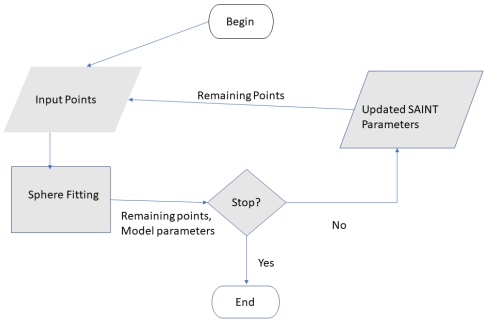

Figure 2. 3D Surface Aggregation in Topology Representation (SAINT)-based modeling Flowchart.

3.2. 3DSAINT-based shape fitting

Model fitting of the sphere in our approach is an optimization problem, where we search for those parameters of the sphere that best fits the initial set of points. Our system utilizes two schemes: *random sampling* and *progressive sampling*. In *random sampling*, spheres are fit to points that best minimize the error by random sampling of 3D points. In *progressive sampling*, 3D data is divided to contain a set of bounding volumes (cubes). Each cube contains 3D points for which spheres are fit, starting with the smallest radius to the largest radius that fits within the bounding cube. We use a parameter to govern the number of spheres per cube. We then end up with a set of sphere centers, radii, and binary *contact maps*, which lead to the best agreement between the selected points and the resultant model. The error of fit assesses this degree of consensus. A lower value indicates a higher degree of match and is governed by a threshold that controls the number of points chosen for that fit. We get a series of *contact maps* based on the number of spheres chosen. All the binary contact maps fuse into one, where a grid location (representing the latitude and longitude of a sphere) would have the corresponding sphere number. We have three floating numbers per sphere representing its center (x, y, z), one floating number per sphere representing the radius, and one merged *contact map* for all the spheres. The coalesced contact map is a sparse matrix with integer numbers (sphere numbers) at various grid locations. The resolution governs the dimension of the *contact map* one would like to achieve in modeling the object, thus representing a multi-resolution representation of the modeled surface.

Although geometric model fitting is fundamental in computer graphics and computer vision, most geometric model-fitting methods cannot fit an arbitrary geometric model to incomplete data. Typically, given a point cloud representation of a shape, prior works estimate the most plausible primitives to fit sequentially. i.e., a given table is represented hierarchically by identifying the primitive that fits the top surface first and then the legs successively [41]. Instead, we search for the best sphere at each stage. We can think of it as querying the model space with our point cloud data for model parameters. Thus, a tabletop (a plane) is modeled with a sphere of a sufficiently large radius. A geometric model is a continuous point set (e.g., a surface), and a geometric model space is a set of geometric models. Our geometric model space contains all the possible spheres with different centers and radii. We thus iteratively retrieve the desired model from randomly selected initial points. After each fit, inliers are recorded through sphere radius, center, and the corresponding *contact map*.

In Figure 3, we illustrate our modeling process. In this figure, (i) represents a set of 3D points. In the first iteration of fitting, we fit a sphere to all the red points and the corresponding polar coordinates are recorded in the contact map, as shown in (ii). We are left with the remaining points, as shown in (iii). Our next iteration would fit the sphere to the blue points and other locations of the contact map had been turned on, as shown in (iv). The remaining points are shown in (v), fitted with another sphere, and contact points are recorded as shown in (vi). The remaining points gradually decrease, as shown in (vii), and finally, the last fit is made in (vii).

We fit spheres to the entire object for a single object, whereas for a scene, we partition the input 3D surface data into regions. We then approximate each region's geometry by spheres such that the final squared distance from the 3D point set to its spheres approximation is minimized. Our key idea is that instead of using a point cloud to represent a 3D model, we use patches or discrete points from spheres to represent a group of points. Furthermore, we emphasize that local neighborhoods on a sphere may or may not translate to neighborhoods in 3D. The raw unstructured 3D points usually lack connectivity to the surfaces they belong to, which we happen to model using spheres. Each point P can be re-created back using ρ, θ and ϕ, available for each point in our contact map(s) (Eq. 1). ρ is the distance from the pole, ϕ is the angle from the z-axis (colatitude, measured from 0 to 180 degrees) and θ is the angle from the x-axis (measured from 0 to 360 degrees).

$$P(\phi, \theta) = (\rho sin(\phi)cos(\theta), \rho sin(\phi)sin(\theta), \rho cos(\phi)) \quad (1)$$

Given the point cloud representing an object (or a scene), we model it as a union of spheres (S), given as below (Eq. 2):

$$P_{Obj1} = S_{11} \cup S_{12} \cup S_{13} \cup ... \quad (2)$$

An object (or a scene) thus is represented as a union of modeled spheres (see Figure 4).

Say we have a scene with three objects and modeled each

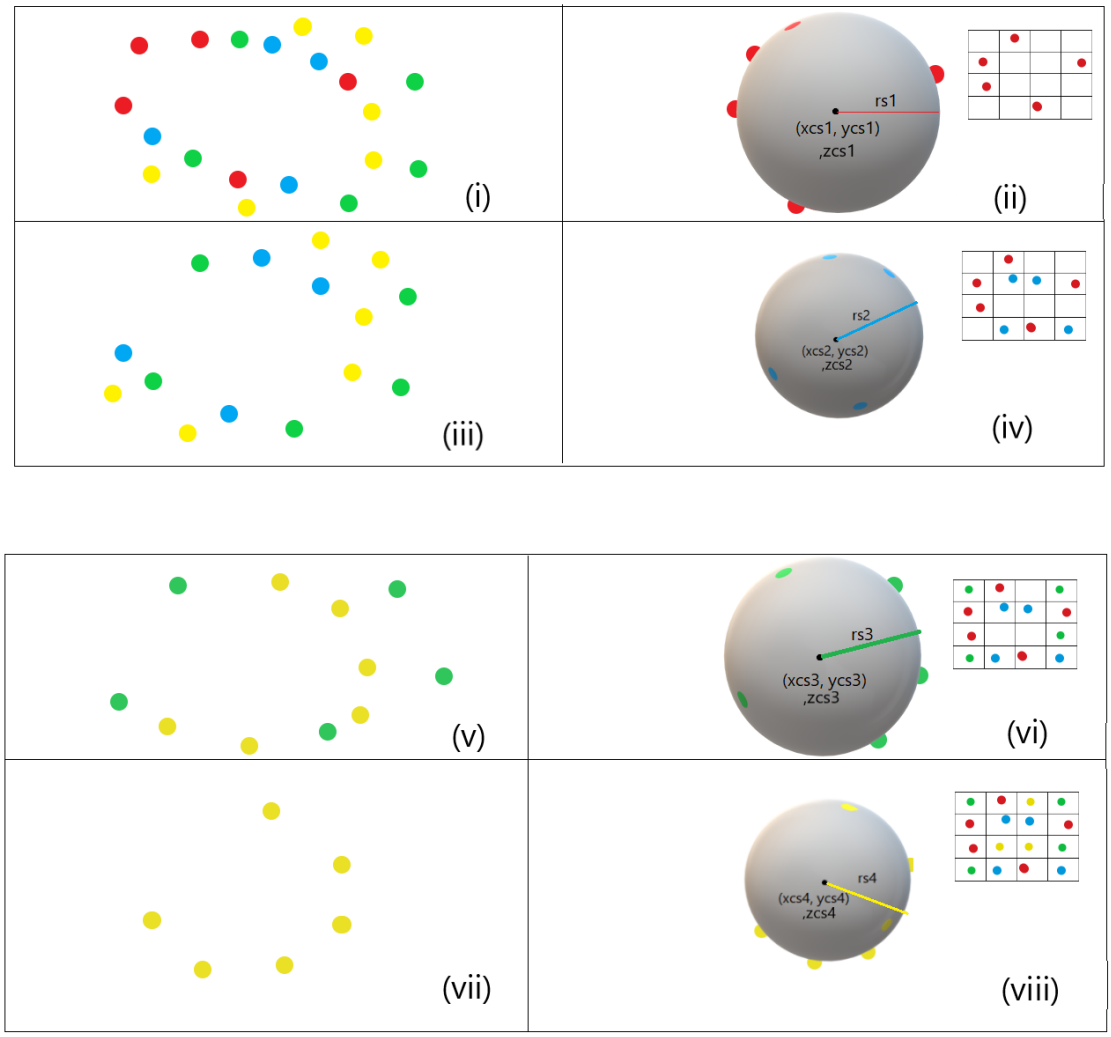

Figure 3. Fitting Procedure: The left column illustrates 3D points; the right column illustrates modeled Spheres, corresponding radii, centers, and contact maps. (i) the 3D points, (ii) First Sphere fit, (iii) Remaining 3D points, (iv) Second Sphere fit, (v) Remaining points, (vi) third Sphere fit, (vii) Remaining points, (viii) fourth Sphere fit.

object with three spheres. Then,

$$P_{Scene1} = P_{Obj1} \cup P_{Obj2} \cup P_{obj3} = S_{11} \cup S_{12} \cup S_{13} \cup S_{21} \cup S_{22} \cup S_{23} \cup S_{31} \cup S_{32} \cup S_{33} \quad (3)$$

4. Experiments

4.1. Modeling and Storage

We present our experimental results on diverse 3D data: face, teeth, and kidney. For the face, we use the Bosphorus Database [28], which consists of facial data acquired using a structured-light-based 3D system. 105 subjects under various poses, expressions, and occlusion conditions are available, with the total number of face scans being 4666. The 3DSAINT-based modeling involves setting two main parameters: the number of spheres to use and the distance threshold of points from the sphere to consider as a contact (*contact distance*). Once the *contact distance* is set and the number of spheres increased, we can obtain the 3DSAINT-based representations with an increasing number of 3D points modeled (see Fig. 5). For each set number of spheres, if the *contact distance* is varied, we achieve multi-scale representation of the data being modeled. Fig. 7 represents three scales of the face achieved using 5, 50, and 200 spheres, respectively. *Contact distance* is different for each of the configurations. In this figure, each color within

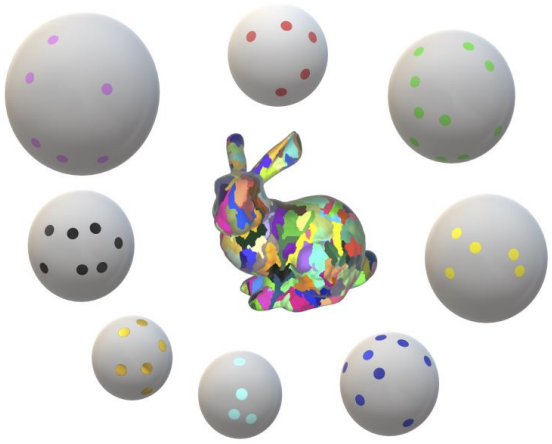

Figure 4. Single Object represented as Union of Sphere points. Each different sphere models regions on the surface with corresponding contact points.

Figure 5. 3DSAINT-based modeling with number of Spheres fit: 10, 20, 50, 100, 200 and 250.

the face corresponds to a particular sphere being *contacted* for those points. Finer details are achieved as the number of spheres used for 3DSAINT-based modeling increases.

Figure 6 consists of CT imaged teeth on the top left, 3DSAINT-based reconstructed teeth as 3D points, and the original teeth overlayed with 3DSAINT-based reconstructed 3D teeth points. We can see that 3DSAINT-based performs a faithful reconstruction of the teeth data. We further used these results to visualize in a Mixed Reality setup. SAINT can represent the data using randomly sampled points to fit each iteration for generating inliers/outliers set (such as in RANSAC) (*random method*), or done in an organized way, which we call it *progressive method*. We divided the 3D space into volumetric regions and fit spheres from small to large in the *progressive method*, 25 per cube. We now describe storage results on both these methods.

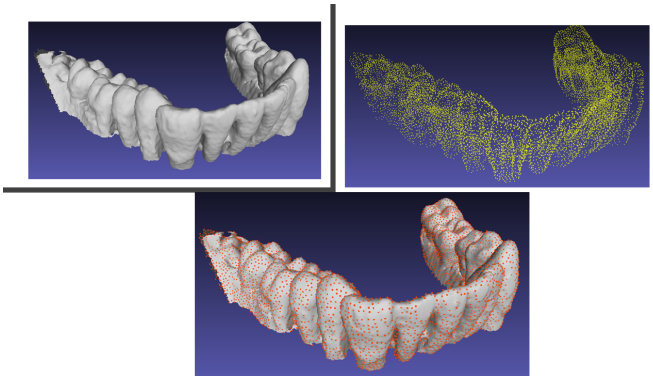

Figure 6. Dental 3DSAINT-based result.

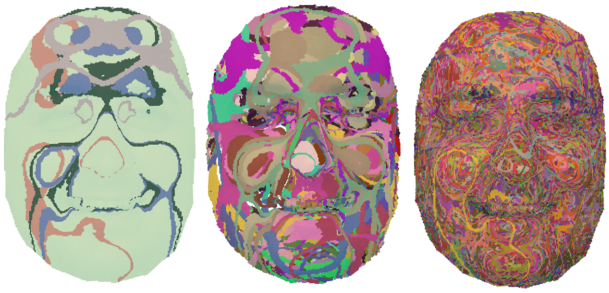

Figure 7. 3DSAINT-based faces at increasing scales. Colors represent each sphere, and spheres are increased from left to right.

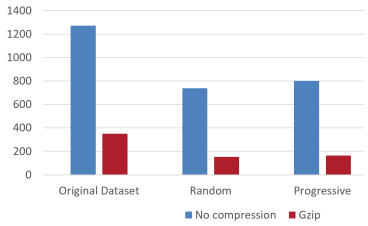

Figure 8. 3DSAINT-based face storage in Megabytes.

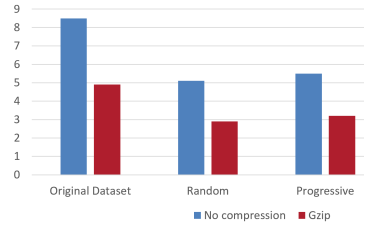

Figure 9. 3DSAINT-based ScanNet storage result in Megabytes.

One of the main advantages of our 3DSAINT modeling is its storage efficiency. We have performed numerous experiments on the storage efficiency of the 3DSAINT-based representation compared to a general compression mechanism. Figure 8 shows our results on the 3D face dataset

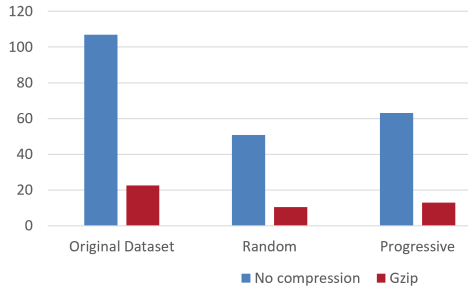

Figure 10. 3DSAINT-based Kidney storage result in Megabytes.

(Bosphorus Database) and Figure 9 shows our results on the ScanNet data [5]. Figure 10 shows our results on the kidney dataset, which is from Kidney Tumor Segmentation (KiTS19) Challenge [9]. The data consists of 210 patients, where physicians provided manual segmentation labels. The 3D surface of the kidney was created using the Marching cubes algorithm. It is clear from these figures that 3DSAINT-based representation has considerably decreased the storage requirements. Note that Gzip of original data and then Gzip of 3DSAINT-based representation data show a considerable increase in compression.

Johannes Kepler proposed that the optimum way of sphere packing as densely as it can be is to pile them as one would see in a grocery market with oranges or tomatoes (see Fig. 11(c)). This is popularly termed as Kepler's Conjecture [31]. This packing method is better than a regular grid, such as in 11(a). Thus we use the 3D grid version of 11(b) to pack spheres in a given object or scene. We thus divide the entire 3D space into volumetric regions and initialize spheres at the center of these volume regions. Figure 11(d) shows one of the original images from ScanNet, Figure 11(e) shows the residual error with regular packing and Figure 11(f) shows residual error for hexagonal packing, clearly showing reduced residual error. Figures 12, 13, and 14, show the ground truth, our results on *random* and *progressive* versions of our 3DSAINT-based method, on a sample data respectively. We found little difference except for a slight pattern in the organized fitted model. In Figure 15, we present our reconstruction error, where we obtained the average error of all objects in ModelNet40, modeled by 3DSAINT using the Chamfer Distance compared to the ground truth; the percentage error is reduced as the contact map resolution is increased.

4.2. Classification, and Structure from Motion

We use 3D SAINT representation for classification to show improvement to the pointnet++ architecture [25], RepSurf [27], and PointNeXt [26]; our results are presented in Figure 16 and Figure 17. We add features from our 3D SAINT representation that includes the sphere center,

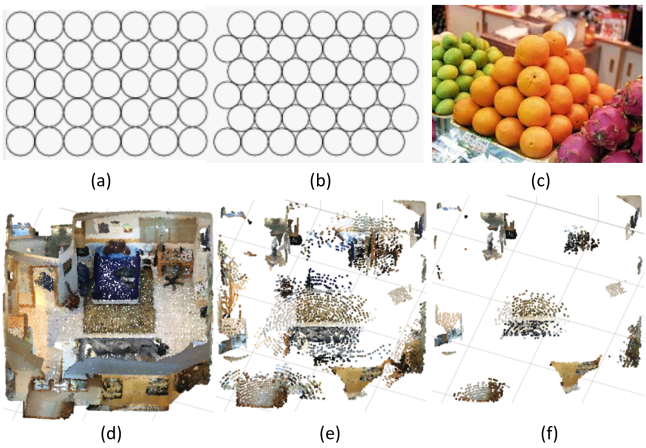

Figure 11. Packing Configuration.

Figure 12. Original Scene from ScanNet.

radius, length of the vector projected onto the xy-plane, and the angle between the projected vector and the x-axis. The improvement is considerable (ranging from 0.3-0.4) on ModelNet40 shown in Figure 16, and ScanObjectNN (ranging 0.5-0.6) shown in Figure 17.

We have incorporated our system into Structure from Motion frameworks, TransDepth [39], AdaBins [2], P3Depth [22], and SwinV2-B 1K-MIM [37]. The loss function in these frameworks are modified using the 3DSAINT loss, incorporating the distance from the spheres modeled. As seen in Figure 18, the reconstruction error is decreased by using our method.

5. Conclusions

There is considerable interest in compressed and compact representation for storing and retrieving effective 3D

Figure 13. Reconstruction of the Scene based on Random 3DSAINT.

Figure 14. Reconstruction of the Scene based on Progressive 3DSAINT.

content generated by modern 3D sensors. In this paper, we approximate shapes with simple geometric primitives, spheres. We aim to represent the input 3D object with few parameters for efficient storage, transmission, and recognition. As a result, our storage is reduced considerably, which contains sphere radii, centers, and a unified *contact map*. We utilize a contact map array to record the region of shape that is close to each sphere and use sparse representation for this. Our method successfully represents the input shape with spheres with minimal input parameters. Our storage is one-third of what a standard compression scheme obtains. Also, Spherical-based representation encodes 3D data such

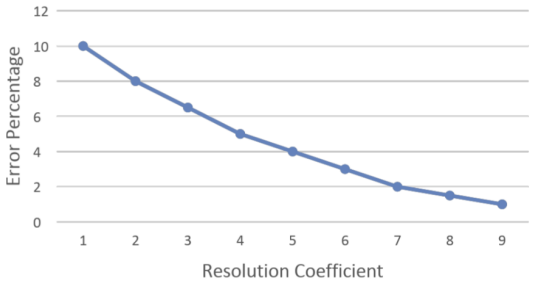

Figure 15. Reconstruction Error with contact map resolution.

Method	Original	SAINT
PointNet++	90.7	**91.1** (+0.4)
PointNet++ (with normals)	91.9	**92.2** (+0.3)
RepSurf	93.2	**93.5** (+0.3)
PointNeXt	93.2	**93.6** (+0.4)

Figure 16. 3DSAINT-based Classification Improvement on ModelNet40.

Method	Original	SAINT
PointNet++	77.9	**78.4** (+0.5)
RepSurf	85.5	**86.0** (+0.5)
PointNeXt	87.7	**88.3** (+0.6)

Figure 17. 3DSAINT-based Classification Improvement on ScanObjectNN.

Method	RMSE
TransDepth	0.365
AdaBins	0.364
P3Depth	0.356
SwinV2-B 1K-MIM	0.304
SwinV2-B 1K-MIM + SAINT	**0.301**

Figure 18. Structure from Motion improvement results.

that it is hard to decode the data without knowing some crucial parameters, thus protecting the 3D point data from Cyberattack. Our method facilitates multi-scale representation, which will be helpful for many tasks in Mixed Reality and computer vision.

References

[1] Draco 3d data compression, google. https://google.github.io/draco/. 3

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 7

[3] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. 1

[4] Hui Cao, Haikuan Du, Siyu Zhang, and Shen Cai. Inspherenet: A concise representation and classification method for 3d object. In Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve, editors, *MultiMedia Modeling - 26th International Conference, MMM 2020, Daejeon, South Korea, January 5-8, 2020, Proceedings, Part II*, volume 11962 of *Lecture Notes in Computer Science*, pages 327–339. Springer, 2020. 3

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7

[6] M. Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH '97*, 1997. 3

[7] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *ECCV*, 2018. 2

[8] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *CoRR*, abs/1912.12033, 2019. 2

[9] Nicholas Heller, Niranjan Sathianathen, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, Joshua Dean, Michael Tradewell, Aneri Shah, Resha Tejpaul, Zachary Edgerton, Matthew Peterson, Shaneabbas Raza, Subodh Regmi, Nikolaos Papanikolopoulos, and Christopher Weight. The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes, 2020. 7

[10] Long-Nhat Ho, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Toward realistic single-view 3d object reconstruction with unsupervised learning from multiple images. 2021. 3

[11] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys*, 50, 06 2017. 1

[12] Marc Janeras, Joan Roca, Josep A. Gili, Oriol Pedraza, Gerald Magnusson, M. Amparo Núñez-Andrés, and Kathryn Franklin. Using mixed reality for the visualization and dissemination of complex 3d models in geosciencesmdash;application to the montserrat massif (spain). *Geosciences*, 12(10), 2022. 1

[13] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 2

[14] D. G. Kendall. *Shape and Shape Theory*. Wiley, 1999. 3

[15] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. *CoRR*, abs/1705.05548, 2017. 2

[16] Keith Kirkpatrick. 3d sensors provide security, better games. *Communications of the ACM*, 61:15–17, 05 2018. 1

[17] Robert L. Larkins, Michael J. Cree, and Adrian A. Dorrington. Analysis of binning of normals for spherical harmonic cross-correlation. In Atilla M. Baskurt and Robert Sitnik, editors, *Three-Dimensional Image Processing (3DIP) and Applications II*. SPIE, feb 2012. 2

[18] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3d point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1297–1304, June 2006. 2

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[20] Francisco-Angel Moreno, José-Antonio Merchán-Baeza, Manuel González-Sánchez, Javier Gonzalez-Jimenez, and Antonio Cuesta-Vargas. Experimental validation of depth cameras for the parameterization of functional balance of patients in clinical tests. *Sensors*, 17(2), feb 2017. 2

[21] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M. Cortelazzo. *Microsoft Kinect™ Range Camera*, pages 33–47. Springer US, Boston, MA, 2012. 2

[22] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022. 7

[23] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. *arXiv preprint arXiv:1904.09664*, 2019. 2

[24] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 2

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 2, 7

[26] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS*, 2022. 7

[27] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022. 7

[28] Arman Savran, Nese Alyuz, Hamdi Dibeklioglu, Oya Celiktutan, Berk Gokberk, Bulent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. pages 47–56, 01 2008. 5

[29] Daniel Schmitter, Pablo García-Amorena, and Michael Unser. Smoothly deformable spheres: Modeling, deformation, and interaction. In *SIGGRAPH ASIA 2016 Technical Briefs*, SA '16, New York, NY, USA, 2016. Association for Computing Machinery. 1

[30] Richard Szeliski. Computer vision algorithms and applications, 2020. 1

[31] George G. Szpiro. Kepler's conjecture: How some of the greatest minds in history helped solve one of the oldest math problems in the world. 7

[32] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Om Sai Ram*, July 2017. 3

[33] Thang Vu, Kookhoi Kim, Haeyong Kang, Xuan Thanh Nguyen, Tung M. Luu, and Chang D. Yoo. Sphererpn: Learning spheres for high-quality region proposals on 3d point clouds object detection. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3173–3177, 2021. 2

[34] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR*, abs/1504.03504, 2015. 2

[35] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets for 2.5d object recognition and next-best-view prediction. *CoRR*, abs/1406.5670, 2014. 2

[36] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 160–176, Cham, 2016. Springer International Publishing. 2

[37] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. *arXiv preprint arXiv:2205.13543*, 2022. 7

[38] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. 3d object dense reconstruction from a single depth view. *CoRR*, abs/1802.00411, 2018. 2

[39] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer vision*, pages 16269–16279, 2021. 7

[40] Yuqi Yang, Shilin Liu, Hao Pan, Yang Liu, and Xin Tong. Pfcnn: Convolutional neural networks on 3d surfaces using parallel frames. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[41] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 4