Graphical Abstract

Spherical Harmonics for Robust Next-best-view Estimation

Alexandru POP, Levente TAMAS

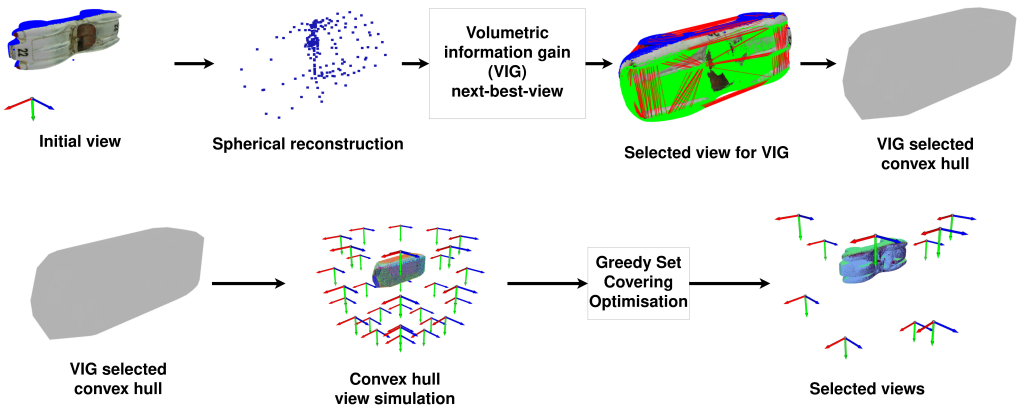


Figure 1: Our proposed next-best-view(NBV) for volumetric information gain (VIG) score and usage of resulting convex hull for efficient coverage reconstruction NBV

Highlights

Spherical Harmonics for Robust Next-best-view Estimation

Alexandru POP, Levente TAMAS

- A preprocessing using spherical harmonics for point-clouds
- A new volumetric score applied directly on 3D point clouds
- An n-step NBV extension that is robust to changes in view space

Spherical Harmonics for Robust Next-best-view Estimation

Alexandru POP, Levente TAMAS*

Abstract

As 3D depth cameras have become more affordable, the extra depth information proves useful in industrial setups. However, with a multiview camera configuration, the complete coverage of a 3D model can only be accurately captured from several views. For this task, the next-best-view (NBV) estimation is essential to minimize the number of required captions for a complete reconstruction. Traditional NBV methods for 3D data are often affected by sensitivity to noise and are not rotation invariant. Learning-based methods can make estimations based on current information from cameras and can significantly improve the accuracy of a task. These methods are generally limited to one step as multistep planning requires the estimation of new sensor observations. A method that can predict and simulate multiple steps can significantly improve performance where one step is insufficient. We show that a convex hull of the object can be used to predict n-steps NBV in a coverage reconstruction setting. The convex hull is used as a proxy for the real geometry of the object in the selection of the views and requires much fewer

*Corresponding author
Email address: Levente.Tamas@aut.utcluj.ro (Levente TAMAS)

points sampled from the ground truth geometry. We propose a volumetric information gain (VIG) score that measures the similarity between the convex hulls obtained from the partial point cloud from selected views and the ground truth point cloud densely sampled from the geometry. We retrain a deep-learning NBV network to predict the views that lead to a point cloud with the highest VIG score and show that spherical harmonics preprocessing can make the network faster. Once the views for the convex hull have been selected, we show that spherical reconstruction of the partial point clouds can further compress the relevant geometric information into a much smaller set of coefficients. This method is robust to changes in the view space, as shown in the tests applied on Shapenet models along with more complex synthetic shapes from HomebrewDB [30], Stanford 3D Scanning Repository [12], and Linemod. [24].

Keywords: pose estimation, spectral analysis, planning

---

1. Introduction

In modern industrial setups, the use of 3D data become widespread in the last years [58]. This is mainly due to the affordable and information-rich characteristics of these cameras, which can also be easily integrated into custom production lines [4]. The 3D data obtained from a camera capture describe a partial geometry that is used for various tasks, such as path planning decisions. Depth cameras lead to a vision-based measurement in which the 3D model extracted from the 3D points is compared with a reference

model [44] or a score is computed that reflects a characteristic of the entire object [3]. One of the main advantages of vision-based measurement is that it does not involve contact with objects and uses cameras connected to a computer decision-making system. 3D cameras can greatly help in obtaining the object geometry and performing classification. A single 3D image is usually enough to perform the classification, but for modeling large objects as described in [5], multiple view captures are needed to obtain a complete 3D reconstruction. Object reconstruction is performed using different camera views as described in Next-best-view applications [71, 38, 28] or by using a geometric prior on an incomplete input to predict the full geometry as described in point cloud completion applications [67, 74, 31, 8, 7, 60, 32]. An interesting approach for using prior geometrical information is to link a partial point cloud to an image-based representation such as Neural Radiance Field (NeRF) as described in [77, 29]. The common theme for all approaches is the usage of incomplete geometry and geometrical priors to estimate missing 3D information.

2. Related work

Deciding on the *next position*, also called *view*, the new information is acquired by using camera methods. Their applicability ranges from scene exploration as described in [23] to complex object reconstruction from multiple views as shown in [5, 27]. Regardless of the sensor used, stereo, lidar, bundle of RGB images [16] or sonar, multiple camera captures, also called views, are

needed to obtain the necessary information to perform the 3D reconstruction of the object or scene.

An alternative approach to multiple-view 3D reconstruction is to complete a partial geometry view from a camera capture or a point cloud. Examples of point cloud completions from partial geometries are described in [76, 36, 67, 74, 31, 8, 7, 60]. A different reconstruction method is to associate direct camera captures with completed geometries, as shown in [74, 46]. This allows for the use of depth images instead of point clouds to directly predict the full geometry of a scene. All these direct reconstruction methods use a priori information about the geometry of the object to fill the empty spaces that the camera would cover from a different position. Another approach is to project the partial point cloud to different views to make useful predictions such as object detection [68] and object retrieval [10]. For complex objects and scenes, the information from a single camera capture is not sufficient to create an accurate reconstruction; however, as more views are gradually added, the amount of missing information is reduced. In applications featuring complex objects, reconstructions made with camera captures guided by a view selection algorithm display more robust results than single-view reconstructions[6].

The points obtained from the camera captures are also affected by different types of noise and need increased robustness to all-season conditions [57]. Practical applications of point cloud processing networks must also take into account adversarial noise, which is position noise specifically designed

to corrupt network predictions, as shown in [16, 62].

This fact underscores the need for a view-predicting method that uses point clouds, is flexible to different types of score, and is highly resistant to different types of noise.

A problem of NBV algorithms is that the next view must be chosen for a region that is not known a priori, leading to difficulties in predicting multiple steps. Current NBV algorithms are designed for the one-step prediction of NBV. In [71], the authors introduced a deep-learning NBV estimation network that uses point clouds as input. The network described in [71], called Point Cloud Next-Best-View($PCNBV$) is relevant to our work as it allows us to link point clouds to custom scores and future view encodings. The original version described in [71] predicts the scores obtained in possible next views for a given incomplete shape point cloud. This suggested setup requires resistance to different types of noise, as perturbations degrade score predictions.

2.1. Next-best-view

The Next-best-view(NBV) estimation is done by maximizing a certain score. The score is usually related to the surface coverage completion of an object [20, 21, 71] but can also be related to scene exploration [23] or different scores like the accuracy of volume estimation [49], image-based neural rendering [29] or plant phenotyping [66] for which multiple views are needed. Object reconstruction from images is related to NBV algorithms, as a prior

capture of an object is used to predict the shape and associated score, as presented in [9, 53]. In [15], the authors show this connection using a point cloud completion module to estimate unknown points from different views and using them to compute a coverage score. Reconstruction from multiple views requires the estimation of the pose of the camera and the registration of images [18, 56], which call for additional methods in a real environment to minimize the effect of noise [6, 25, 48].

A key factor in NBV selection is the method for computing the score that is used to rank views. The score represents the amount of information that can be extracted from a candidate's view. As the NBV algorithm does not have access to the points of the candidate views, it usually estimates information related to the next view and uses it to compute the view score. In [28], the score is called the *Volumetric Information Gain(VIG)*, and it conveys the amount of unexplored space from the candidate's view. Volumetric scores usually use a 3D voxel representation to determine the amount of new information in a view. We implemented a novel volumetric score that is applied directly to point clouds using the volume computed from a convex hull. This new score links the point cloud shape to the volume it describes and the relative difference from the original shape volume. A volumetric score is closely related to the shape described by the point cloud. In a multi-step NBV setting, the point cloud after the next step is not available, and only a rough prediction of the shape can be used. This means that the actual point clouds from the views can be used to compute ground-truth scores but

a method is needed to link a predicted point cloud shape to the ground-truth scores.

Traditional NBV methods [13] have a generate and test approach in which a 3D space is divided into known, empty, and unknown regions and a ray tracing algorithm is applied from each candidate view location to determine the number of unknown regions that are converted into known or empty.

Newer methods use deep [71, 38, 22, 20, 21] or reinforcement learning [47, 61, 11] to avoid the computationally intensive ray tracing and generation process by finding a link between the current available geometry and the scores obtained in future views, in a one-step horizon. Learning is also applied for one-shot methods [26, 41, 42] to link the current geometry with the collection of views that would solve the set-coverage optimization (SCO). An alternative to predicting scores using learning is to use the currently available geometry obtained from the partial point cloud P_{par} to complete the shape to fill in the gaps and obtain a rough complete 3D model which is used to simulate the results obtained from each candidate view. The point cloud generation is used in [14, 15, 37] in which a partial point cloud is used to generate a predicted point cloud after each view. A mesh generation approach is used in [43] where an input image is used in a mesh generation algorithm and the generated mesh is used to predict the one-shot NBV views.

The authors in [23] have demonstrated a learning method that uses as input 3D voxels that describe a shape and outputs a volumetric score associated with that shape. Their work implies that a network trained in point-cloud

encoding can predict an associated ground-truth volumetric score. Standard point cloud classification networks [50, 51, 64] display the link between a partial point cloud, a corresponding feature encoding, and a final classification score. Learning networks have been successful in various stages in regressing the score of encoding in the point cloud [38, 59, 71]. The link between the point cloud and score allows the evaluation of different views, but it does not offer useful information to predict the information gained after more than one step.

A stable benchmark for learning NBV is the Point Cloud Next Best View ($PCNBV$) network [71]. It treats the NBV as a classification problem from a fixed selection of views and uses P_{par} and the viewstate V_n for n views. Further works use it as a baseline in different contexts such as iterative using learning [21, 20, 22], one-shot NBV [26, 42] or reinforcement learning [61, 11]. Iterative methods aim for higher overall surface coverage. One-shot methods aim for fewer spaces and fewer required views to obtain a similar or better reconstruction whereas reinforcement learning aims for a generalizable view selection strategy with a much larger view space [11].

A relevant observation from [42] is that the coverage gains follow a power law, with the first views contributing most of the coverage. Because of this, later views do not change greatly the object geometry, leading to more stable one-shot predictions if the object is already covered by the first views. Thus a combination between an iterative method like $PCNBV$ and a one-shot prediction can obtain better performance than each one separately.

Another relevant observation discussed in [11] is that a fixed view space achieves a lower surface coverage than the one obtained from a much larger view space. A reinforcement learning approach can surface the performance of a network trained on a fixed sampling of views such as *PCNBV* simply by having access to better views.

Our idea is similar to [43], but instead of generating a mesh from an image, we propose first to construct a convex hull for the object of interest using selected views and then to apply set-coverage optimization (SCO) with a simulated depth camera for the convex hull. We use the intuition of [42] that an approximate model built from initial views can greatly help the one-shot prediction. Considering a low sampling of points from each random view, the resulting shape is a very rough approximation of the ground-truth geometry. Our claim is that this approximate convex hull can be successfully used to perform a one-shot NBV prediction. The direct advantage of this method is that it is not limited to a predefined set of views, and once the convex hull is obtained, any view can be simulated and compared.

Point cloud processing networks are sensitive to noise, such as Gaussian position noise, rotation noise, and adversarial noise, as explained in [40, 65]. Adversarial attack noise is a particular class of position noise that is added to an input point cloud in order to alter the predictions of the targeted network. In [55], the authors showed that single-view learning networks are particularly vulnerable to adversarial attacks. In [1, 2, 39, 40], the authors have highlighted that point cloud processing networks employing spherical harmonics

have improved resistance to noise. In [39], the authors have performed extensive noise tests using spherical harmonics and discovered a higher resistance to noise than DGCNN [64] and PointNet++[51]. Deep learning methods applied to spectral data have been very promising in other complex signal processing classification tasks, such as mental state decoding using EEG signals [34].

2.2. Spherical harmonics

Spherical harmonics present a way to regress a function applied on a set of spherical coordinates [53]. It is an extension of Fourier analysis, where a function applied on a 2D surface in 3D space is approximated with a finite set of coefficients. The points of the mesh or the point cloud are transformed into spherical coordinates on the unit sphere, considering the latitude $\phi \in [0, \pi]$ and longitude $\theta \in [0, 2\pi]$. A function applied to each point can be rewritten as a function applied to the spherical coordinates. The function applied on the spherical coordinates of the points can represent real physical values that must be modeled, such as radiance fields [17, 33, 73] or acoustic fields [59] but it can also represent the norm of each point as was used in [2, 40, 53]. The spherical coordinate function is decomposed into a sum of weighted basis functions. The basis functions are called spherical harmonics and the corresponding weighting coefficients for each are called spectral coefficients.

Knowing the coordinates and values of the function in each point, the parameters of an approximating function can be regressed, obtaining a link

between an input shape and the corresponding spectral coefficients. This method was used in [45] to determine the shape of the free space available near the sensor, which allows a robot to navigate an environment without collisions. In [35], a connection was made between the shape of the models and the representation of the spherical harmonic, allowing the generation of new synthetic models and the estimation of the spherical harmonics for the real models. This supports intuition from [53], where new spherical signatures were generated in a generative adversarial setting. A similar approach is described in [52], where the spherical harmonics were determined for different grain shapes. In [63], the authors also presented the usage of spherical harmonics to generate new shapes for a class of objects. In [19], the authors showed a connection between shape and spherical harmonics to determine cranial deformations. As noted in [45], a subsampled set of key points can have a similar spherical harmonics signature with the source point cloud even if the number of points is orders of magnitude lower.

2.3. Contribution

The proposed method establishes a link between a convex hull and a one-shot NBV prediction which is adaptable to different view spaces and can be used to plan n-steps. Additionally, we demonstrate that a point cloud completion network can be used in a one-shot NBV estimation context by using the convex hull of the predicted point cloud. A visualization of the proposed method is seen in Fig. 2. Our contributions thus are three-fold:

1. A point cloud processing method using spherical harmonics
2. A new volumetric score applied directly on 3D point clouds
3. An n-step NBV extension that is robust to changes in view space

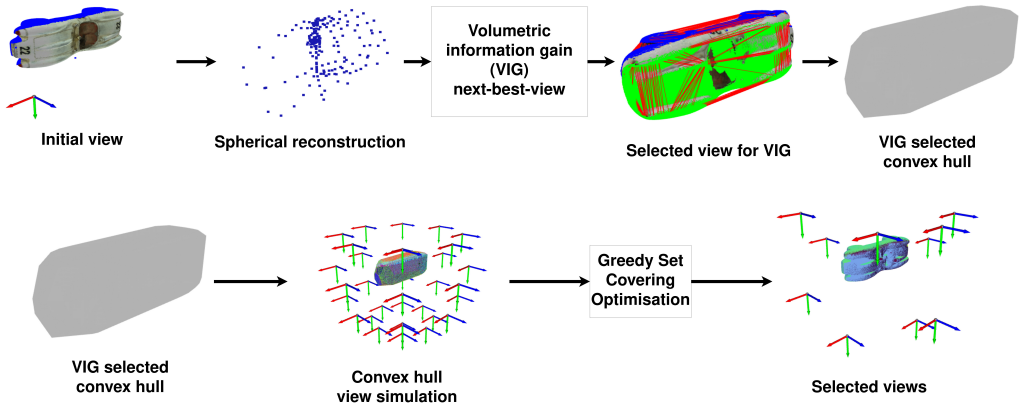


Figure 2: Our proposed next-best-view(NBV) for volumetric information gain (VIG) score and usage of resulting convex hull for efficient coverage reconstruction NBV

3. Theoretical foundations

3.1. NBV

Starting with a depth camera and an *object of interest* $\mathcal{O}$, by moving the camera to a new position and orienting the camera towards the object a point cloud P is obtained. Considering a bounded 3D space $V \in \mathcal{R}^3$ where the camera can be placed, a sampling of $k \in \mathcal{N}$ valid *camera views* $v_k \in V$ is chosen. Each camera view pointed towards the object O leads to an associated depth image. A point cloud for the view $P(v_k)$ is used. The point coordinates are according to the depth camera position and orientation, thus

a registration function f_{reg} is needed to bring the point clouds $P(v_k)$ in the same coordinates. The object of interest $\mathcal{O}$ is approximated as a surface in 3D space which is best described as a mesh $M_{\mathcal{O}}$. The registered point clouds $P_r(v_k) = f_{reg}(P(v_k))$ are thus a sampling of points from the object mesh $M_{\mathcal{O}}$. As more points from different views are added, the resulting point cloud starts covering the surface of the object mesh $M_{\mathcal{O}}$. If the object geometry is known, then a sampling of any number of points is possible from the object mesh $M_{\mathcal{O}}$ leading to the object point cloud $P_{\mathcal{O}}$. In real camera applications, the underlying geometry of the object is not known but a sampling of a large number of points can be obtained from concatenating many depth images from all around the object leading to the associated ground truth point cloud $P_{\mathcal{O}}$. Using a concatenation function $\oplus$ the points from all the views are added leading to a point cloud very similar to the ground truth sampled one, as described in (1).

$$\oplus_{k=1}^{n}(P_r(v_k))) \rightarrow P_{\mathcal{O}} \tag{1}$$

Thus the collection of registered point clouds and the ground truth can be compared. Any selection of views can be codified into a viewstate vector $V_{state} \in \mathcal{R}^n$ as shown below in (2).

$$V_{state}(i) = \begin{cases} 1, & \text{if } v_i \text{ visited} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The viewstate V_{state} also indicates the positions which have not been visited. Knowing the viewstate, a partial point cloud $P_{par}(V_{state})$ is computed as described in (3).

$$P_{par}(V_{state}) = \oplus_{i=1}^{n}(P_r(v_k)) \cdot V_{state}(i)) \tag{3}$$

A utility function $f_{util}(P_{par}(V_{state}, P_O))$ is defined to compare the partial point cloud to the ground truth. The long-term goal of the NBV algorithm is to use a starting view v_i and viewstate V_{start} and find a sequence of m views v_k with $m < n$ that leads to a viewstate V_m such that $f_{util}(P_{par}(V_m), P_{\mathcal{O}})$ has maximum value.

To achieve this goal, the common method is to use a greedy approach where the partial point cloud is used to determine the next view from the unvisited ones which will lead to the partial point cloud with the highest score. Considering the function h which updates the viewstate V_{state} with the view v_i, the *next view score* prediction function $NVS(P_{par}(V))$ is thus

$$NVS(P_{par}(V_{state})) = \begin{bmatrix} f_{util}(P_{par}(h(V_{state}, v_1)) \\ f_{util}(P_{par}(h(V_{state}, v_2)) \\ \cdots \\ f_{util}(P_{par}(h(V_{state}, v_n)) \end{bmatrix} \tag{4}$$

After the NBV view is selected using $NBV = argmax(NVS)$, the viewstate is updated $V_{state}(NBV) \leftarrow 1$ and afterwards the partial point cloud is updated $P_{par}(V_{state}) \leftarrow P_{par}(V_{state}) + P_r(v_{NBV})$. Classical NBV algorithms

use the generate and test approach in which the result of $f_{util}(P_{par}(V_{state}) + P_r(v_i))$ is computed by using ray tracing to determine the new information in the selected view. In the setting described in [71], the network is trained to associate the output NVS with the input $P_{par}(V_{state})$. It associates the current geometry of P_{par} with the scores obtained after adding the points in each view. The viewstate is also added as an input to better differentiate between partial point clouds in different stages.

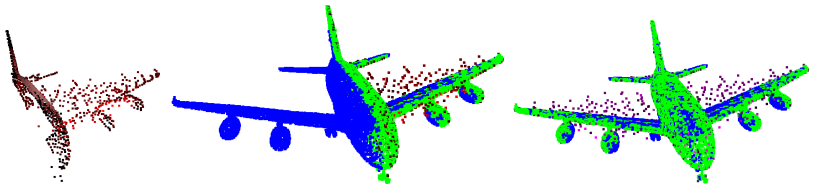
3.2. Surface coverage

The function $f_{util} = Cov(P_{par}(V_{state}), P_{\mathcal{O}}, \varepsilon) \in [0, 1]$ used in [71] is a *coverage function* and it describes how many points from $P_{\mathcal{O}}$ have a neighbor in $P_{par}(V_{state})$ at a distance lower than the threshold $\epsilon \in \mathcal{R}$. The exact mathematical formula for the coverage function is shown in (5).

$$Cov(P_{par}(V_{state}), P_{\mathcal{O}}, \varepsilon) = \frac{1}{|P_{\mathcal{O}}|} \sum_{p \in P_{par}} \mathrm{U}\left(\min_{p_o \in P_{\mathcal{O}}} \|p - p_o\|_2 - \varepsilon \right) \tag{5}$$

In [71], the authors used surface coverage using *K-nearest neighbor* on point cloud representation described in eq.(5) and shown in Fig 3.

The surface coverage utility function depends heavily on the choice of $P_{\mathcal{O}}$ and the threshold ε. A small ε requires a $P_{\mathcal{O}}$ with many more points than a point cloud from a single view $P(v_i)$. The uncertainty regarding the threshold and the size of the point cloud $P_{\mathcal{O}}$ can increase the difficulty in applying the algorithm to unknown object geometries with varying sizes.



(a) Point cloud view 1 (b) Coverage for an initial view (c) Coverage after NBV move

Figure 3: Surface coverage example using densely sampled ground truth point cloud (blue), partial point cloud (red) and selected points (green)

3.2.1. Volumetric Information Gain

For any point cloud P, a *convex hull* M_P can be determined, and the volume Vol_P. Thus as $P_{par} \rightarrow P_{\mathcal{O}}$, the associated convex hull also converges as $M_{P_{par}} \rightarrow M_{P_{\mathcal{O}}}$ leading also to $Vol_{P_{par}} \rightarrow Vol_{P_{\mathcal{O}}}$. Our proposed *volumetric information gain function* is shown in eq.(6).

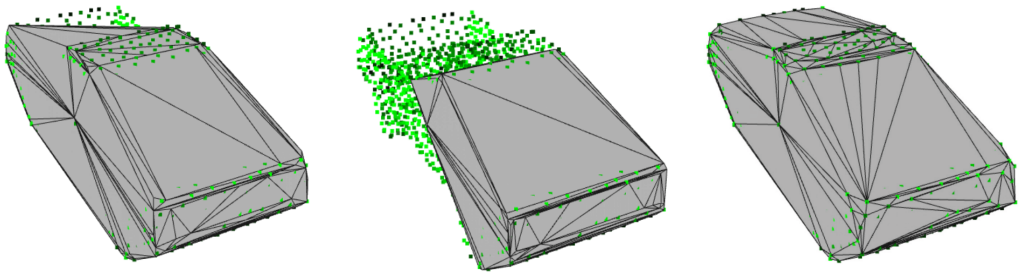
$$VIG(P_{par}(V_{state}), P_{\mathcal{O}}) = \frac{Vol_{P_{par}(V_{state})}}{Vol_{P_{\mathcal{O}}}} \tag{6}$$

The first advantage of this new score is that it requires significantly fewer points sampled from the object to achieve a similar volume. For coverage scores, the ground truth point cloud has 16000 points. We discovered that 512 points are sufficient to obtain a convex hull with a similar volume to the one obtained from the coverage sampling as $Vol_{P_{\mathcal{O}_{512}}} \approx Vol_{P_{\mathcal{O}_{16000}}}$. We discovered that even three random views with 100 points each are sufficient to obtain a convex hull that can be successfully used in the NBV prediction.

The second advantage of the convex hull is that it offers a shape prior with fewer points. For a complex shape with only a part of the 3D geometry available, a point cloud completion approach might fail to describe all the

details of the object but the resulting convex hull can be useful for the NBV prediction.

The convex hull is computed using the Qhull algorithm applied in the Open3D library [75]. We compute the convex hull of the input point clouds and compare their volume to the volume of the convex hull of the ground-truth(GT) object. More points available in describing the model lead to better convex hull results. From the convex hulls, one can compute their volumes and determine what percentage of the GT volume the current point cloud achieves. This procedure is visualized in Fig.4. We used synthetic models to generate the point clouds from each view and compute ground-truth coverage and VIG scores.



(a) Convex hull for GT points

(b) Convex hull hull for points after view 1

(c) Convex hull hull for points after view 2

Figure 4: Convex hull estimation from different point clouds. The sampled points from the object are used in (a) to create the convex hull. The meshes obtained from viewpoint clouds do not cover all the ground truth points, as shown in (b), and (c). The shape of the mesh from a view (b) approaches the ground-truth mesh (a) as more representative points are added (c)

3.3. Spherical harmonics preprocessing

The second improvement is in increasing the speed and robustness to noise of the $PCNBV$ network trained on the new score. Our solution is to process the point cloud using *spherical harmonics* to reduce its size and make it resistant to different types of noise. Applying the *spherical signature* computation function g to $P_{par}(V_{state})$ results in the spherical signature point cloud $P_{sph}(V_{state}) = g(P_{par}(V_{state}))$, where the number of points in the is much smaller than the original point cloud $|P_{sph}(V_{state})| << |P_{par}(V_{state})|$. We trained a $PCNBV$ network[71] to link the spherical signatures and viewstate to the scores obtained after each view.

3.4. Point cloud signature geometry using spherical harmonics

The goal of spherical harmonics is to decompose a complex function on spherical coordinates into a sum of weighted basis functions. We normalized the point clouds in a unit sphere centered in 0 using the center of the points and projected each point coordinate (x_i, y_i, z_i) to the unit sphere to obtain the spherical coordinates (r_i, θ_i, ϕ_i). For any triplet (x_i, y_i, z_i), the conversion to spherical coordinates is done using eq. 7

$$\begin{aligned} r_i &= \sqrt{(x_i^2 + y_i^2 + z_i^2)} \\ \theta_i &= arcsin(z_i/r_i) \\ \phi_i &= atan(y_i/x_i) \end{aligned} \tag{7}$$

Considering $r_i = f(\theta_i, \phi_i)$, spherical harmonics approximate the $f(\theta_i, \phi_i)$ function with a fixed length sum of weighted basis functions. The formula for the spherical harmonics approximation function is given in (8). $l \in \mathbb{N}$ represents the *collection of frequency* bands and $m \in \mathbb{Z}$ represents a selected frequency band. $L_{max} \in \mathbb{N}$ represents the maximum frequency band. For each frequency band, a set of *spectral coefficients* c_l^m and *basis functions* $Y_l^m(\theta, \phi)$ are considered.

$$f(\theta, \phi) = \sum_{l=0}^{L_{max}} \sum_{m=-l}^{l} c_l^m \cdot Y_l^m(\theta, \phi) \tag{8}$$

The *basis functions* can be computed recursively. We used real-valued spherical harmonics with the corresponding formula for each basis function given in (9).

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2} N_l^m \cos(m \cdot \phi) P_l^m(\cos \phi), & \text{if } m > 0 \\ N_l^0 P_l^0(\cos \phi), & \text{if } m = 0, \\ \sqrt{2} N_l^{|m|} \sin(|m| \cdot \phi) P_l^{|m|}(\cos \phi), & \text{if } m < 0 \end{cases} \tag{9}$$

N_l^m represents a *normalization constant* described in eq.(10). P_l^m is an associated *Legendre polynomial*, shown in (11).

$$N_l^m = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}} \tag{10}$$

$$P_l^m(x) = (-1)^m \cdot \frac{(1-x^2)^{\frac{m}{2}}}{2^l \cdot l!} \frac{d^{l+m}}{dx^{l+m}}(x^2-1)^l \tag{11}$$

The number of basis functions can be controlled using a parameter L_{max}. A higher value of L_{max} leads to more basic functions that can better approximate the function, but involves more computational needs for a larger number of coefficients. A lower number of basis functions offers a rough approximation of the function, missing the fine details.

Since the basis functions are fixed and can be computed recursively for each spherical coordinate, the problem of finding the right coefficients to regress the function can be treated as a *least-squares* optimization problem as shown in (12).

$$E = \min_{c_l^m} \sum_{i=1}^{n} \left(f(\theta_i, \phi_i) - \sum_{l=0}^{N} \sum_{m=-l}^{l} c_l^m Y_l^m(\theta_i, \phi_i) \right)^2 \tag{12}$$

The intuition is that the regressed spectral coefficients are a lossy compression of the point cloud geometry. Once the spectral coefficients c_l^m are available, the conversion from spherical to point cloud $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ is done as described in eq. 13.

$$
\begin{aligned}
\hat{r}_i &= \sum_{l=0}^{N} \sum_{m=-l}^{l} c_l^m Y_l^m \left(\theta_i, \phi_i\right)^2 \\
\hat{z}_i &= sin(\theta_i) \cdot \hat{r}_i \\
\hat{x}_i &= cos(\theta_i) \cdot \hat{r}_i \cdot cos(\phi_i) \\
\hat{y}_i &= cos(\theta_i) \cdot \hat{r}_i \cdot sin(\phi_i)
\end{aligned}
\tag{13}
$$

Algorithm 1 shows a pseudo-code for the reconstruction method using all the previous equations.

Algorithm 1 Convert point cloud to spherical harmonics reconstruction

function RecoPcdSph(P_{in}, l_{max})
    Compute spherical coordinates (r, ρ, θ) using eq. 7
    Compute all N_l^m from eq.10 for $0 \leq l$ and $-l_{max} \leq l_{max}$
    Compute all $Y_l^m(\rho, \theta)$ with eq.9
    Solve least-squares from eq. 12, find spherical coefficients c_l^m
    Sample equiangular $4(l_{max}+1)^2$ points from unit sphere to P_s
    Find spherical coordinates for P_s, (r_s, ρ_s, θ_s) using eq. 7
    Compute all $Y_l^m(\rho_s, \theta_s)$ with eq.9
    Compute $\hat{r}_s = f(\rho_s, \theta_s)$ from eq.8 using c_l^m and $Y_l^m(\rho_s, \theta_s)$
    Convert point cloud, P_{sig} using $(\hat{r}_s, \rho_s, \theta_s)$ and eq. 13
    return P_{sig}
end function

The spherical signatures can be used as a preprocessing of the input point cloud or as an encoding of the output point cloud of interest and we investigate both approaches in this paper. For the first approach, a point cloud processing neural network such as *PCNBV* [71] is trained to use spherical signatures as opposed to original point clouds. For the second approach, a

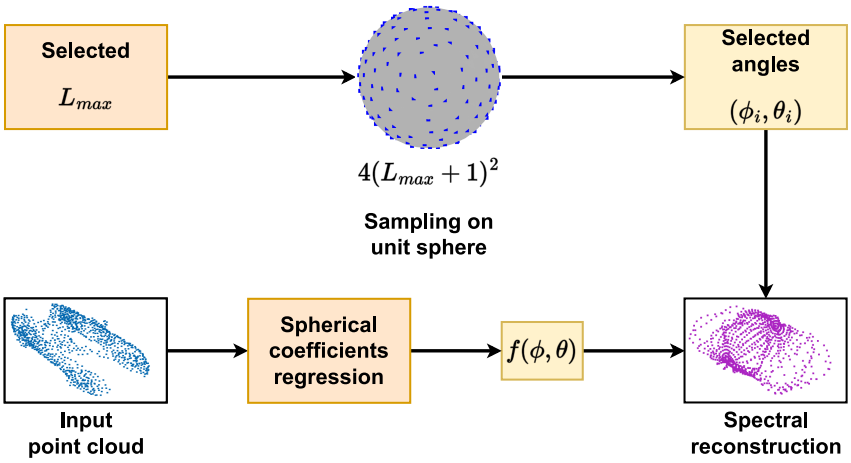


Figure 5: Visualization of the spectral signature reconstruction

partial point cloud P_{par} is used to estimate the spherical signature reconstruction, as shown in Fig. 7.

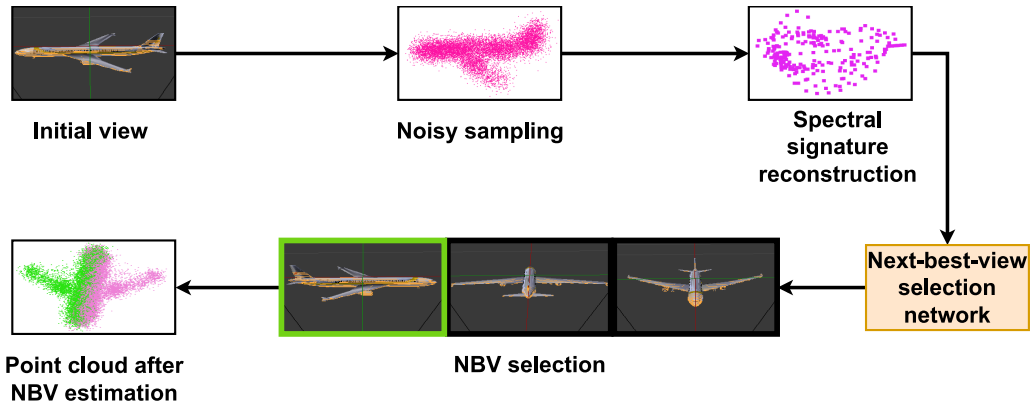


Figure 6: Spherical signature used as a preprocessing and input for a learning based NBV

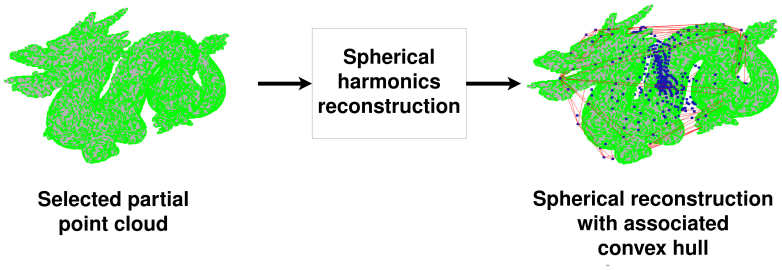


Figure 7: Spherical harmonics reconstruction applied to partial point cloud for a complex object from the Standford dataset.

3.5. Score regression from point cloud

The proposed NBV method uses the *PCNBV* [71] architecture to estimate view scores. The *PCNBV* network shows the ability to regress scores from a point cloud and viewstate. It uses a feature extraction unit from [69] to compute local features and max pooling to aggregate local features into a global signature of the point cloud. As explained in [71], the local and global features are not enough to capture long-range interactions between points; therefore, a self-attention module is used to model spatial dependencies between points at larger distances. Once the self-attention features are computed, a feature processing using shared multi-layered perceptrons is applied, and a max pooling reduces the features to a global signature which is used to select the NBV using a final fully connected layer. The network architecture can be seen in Fig.8.

Since the object geometry is not available, sampling or reconstructing points is necessary. We search for views that lead to a partial point cloud P_{par} for which the convex hull is close to the convex hull of the densely

sampled ground truth point cloud P_{gt}. The comparison between the convex hull of the partial point cloud and the convex hull of the densely sampled ground truth point cloud is encoded in the VIG score.

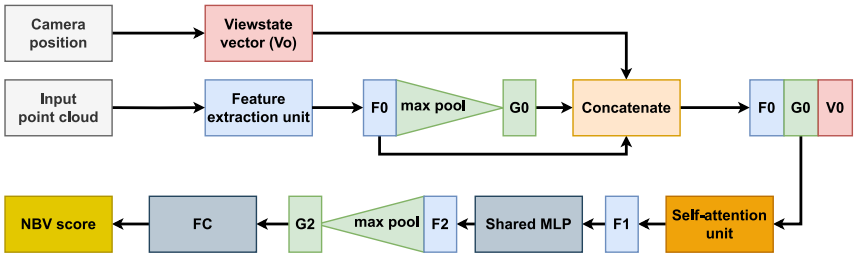


Figure 8: *PCNBV* [71] architecture used to generate spherical coefficients and to evaluate predicted point cloud views. The feature extraction unit proposed in [69] is used to compute local features for each point. The collection of all local features is in F_0. Max pooling is applied to the resulting local features to obtain a global feature G_0. The visited positions of the camera are codified in a viewstate vector V_0. The global feature and the viewstate are copied and concatenated to each local feature. A self-attention module proposed in [72] is used to find long-range relationships between points resulting in the feature vector F_1. Afterward, a convolution using shared multi-layered perceptrons is applied resulting in the feature F_2. Max pool is applied to F_2 resulting in the global features G_2 and a fully connected layer leads to the resulting score value.

3.6. VIG extension for NBV coverage prediction

For each object convex hull M, we simulate views as point clouds $P(v_k)(M)$ and because the convex hull has far fewer triangles than the original 3D model, the simulation is much faster. For a convex hull M, number of views n, viewspace $V_{space} \in \mathbb{R}^{n\times 3}$, and number of densely sampled ground truth points N_{samp}, the coverage can be computed using a set covering optimization (SCO) approach. Densely sampling the convex hull M leads to the ground truth point cloud P_{gt}. Each simulated view from a position $V_{space}(k)$ leads to a point cloud $P_{par}(v_k)$. The points are converted into sets $U(P_{par}(v_k))$

and $U(P_{gt})$. To maintain the same standard as [71], $N_{samp} = 16384$ points are sampled from the convex hull to obtain the point cloud $P_{gt}(M)$. For any view v_k, the set $U(P(v_k)(M))$ is formed by selecting the indices of the points from $P_{gt}(M)$ which have a neighbor from $P(v_k)(M)$ at a distance smaller than the threshold ϵ. We selected the same threshold $\epsilon = 0.00767$ as used in [71] and built the sets for each view. All the sets are added to a list of sets U_{list}. This approach is illustrated in Algorithm 2.

Algorithm 2 Simulate views from input point cloud using convex hull

function SIMCONVHULLVIEWS($P_{in}, V_{space}, n, N_{samp}, \epsilon$)
    $M = ComputeConvHull(P_{in})$
    $P_{gt}(M) = SamplePoints(M, N_{samp})$
    $U_{list} = []$
    for $i = 1$ to n **do**
        $P_{sim} = SimulateView(M, V_{space}[i])$
        $U(P_{sim}) = SelectIndices(P_{gt}, P_{sim}, \epsilon)$
        Append $U(P_{sim})$ to U_{list}
    end for
    return U_{list}
end function

If $U(P_{gt}) \neq \cup_{k=1}^{n} U(P_{par}(v_k))$ then the problem is converted into a Maximum Coverage Problem where the goal is to find a fixed number of sets that form the set with the highest number of elements. To ensure that the SCO can be performed, the ground truth set is considered as the union of all the view sets instead of being formed from the sampling of the original geometry. We did not enforce the ground truth set to be the union of the view sets and kept the ground truth point cloud as a dense sampling from the ground truth geometry, so in our formulation, we have a Maximum Coverage

Problem (MCP). Because some sampled interior points are inaccessible to the views, the union of the views will not cover the entire point cloud. Thus our work leads to results directly comparable to the *PCNBV* scores.

SCO and MCP are NP-hard problems and a greedy approach is the best polynomial-time approximation [54]. The polynomial time ensures the best speed for larger point clouds. For this reason, we used the greedy approach to determine the maximum coverage. We claim that a convex hull allows us to simulate candidate views and select the NBV in a reconstruction setting. The convex hull is used as a proxy for the real geometry in the simulation. The initial views have many points which leads to a rapid elimination of common elements after the first iterations.

We applied the greedy approach to solving the set covering optimization by selecting the views from U_{list} with the most number of points and removing the points from the remaining views. For a fair comparison, we created selections with $n_{iter} = 9$. Algorithm 3 demonstrates the application of the greedy MCP to the sets obtained from the view simulation.

For the coverage reconstruction setting, we compare the spherical harmonics VIG extension with a random selection baseline and a learning method, namely, *PCNBV* [71]. We trained one version of *PCNBV* on partial point clouds and another designated *PCNBV sph* was trained on partial point cloud spherical signatures. We also compare our spherical harmonics VIG extension to [15], which uses prior geometric information to predict the NBV. They train the PoinTr [70] completion network and use it to complete the

Algorithm 3 Greedy Maximum Coverage Selection using Indices

```
function GreedyMaxCovSets(U_list, n_iter, n)
    V_sel = []
    for step = 1 to n_iter do
        v_max = IndexLargestList(U_list)
        Append v_max to V_sel
        for i = 1 to n do
            U_list[i] = U_list[i] \ U_list[v_max]
        end for
    end for
    return V_sel
end function
```

partial point cloud and estimate the contribution from each view. Their work emphasizes the ability to adapt to different views from the fixed view setting. We use the trained PoinTr network from [15] and use their view contribution method on the complex object dataset.

4. Datasets used for evaluation

To compare the NBV learning baseline with the spherical harmonics VIG extension, we used the same protocol as [71], the unseen object test categories being shown in Fig. 9. Additionally, a custom dataset with object models from HomebrewDB [30], Stanford 3D Scanning Repository [12], and Linemod [24] shown in Fig. 10 is used to test the methods for unseen objects. For each object, the views from each position are simulated in Blender with the same method used in [71]. The meshes in the custom dataset are centered using the object center and normalized in a radius sphere 0.5, the same as the models from Shapenet. With access to all views, the NBV methods use a starting

view to obtain 9 next best views in order. By selecting the same initial views for each method, the comparison is done by the coverage obtained after each selected view.

(a) Bed (b) Guitar (c) Bookshelf (d) Motorbike

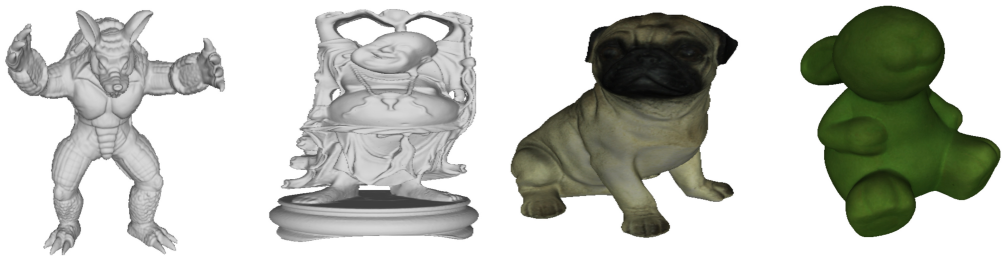
Figure 9: Representative Shapenet unseen test categories

Changing the order and location of the viewpoints is expected to corrupt the predictions of the *PCNBV* network as it is not trained for them. This issue underscores one problem with learning methods for NBV, as the candidate's views must be similar to the ones on which the network has been trained. We use the original viewspace from [71] for training and validating the NBV networks. Afterwards, we modify the viewspace so that the positioning and order of the views are completely different from the original. The original and modified viewspaces are shown in Fig. 11.

5. Results

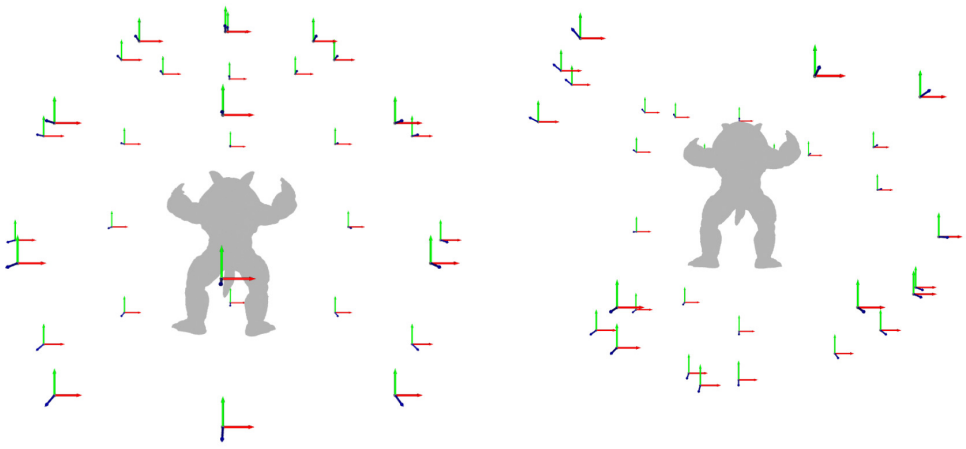
5.1. K-fold validation on PCNBV *training*

To test the prediction ability of the *PCNBV* network, we adapted the Shapenet training dataset to perform a k-fold validation. Since the training



(a) Armadillo (b) Happy Buddha (c) Obj 23 (d) Bunny

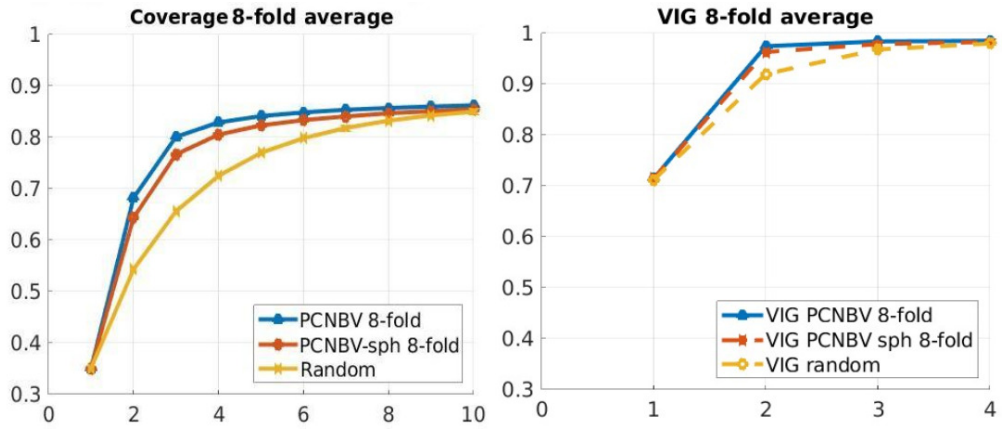
Figure 10: Representative examples of objects from complex geometry



(a) Viewspace used in [71] (b) Changed viewspace

Figure 11: Visualization of original and changed viewspaces. We changed the position and order of the camera locations from the original viewspace so that the new views are from an unseen distribution.

and validation datasets use the same object eight object categories, an 8-fold validation set is constructed by iteratively selecting 7 categories for training and 1 category for validation. We consider the scores resulting from the selection of views to be the most important metric in a NBV context. We test each network on a reconstruction task for each selected category in the validation set and average the scores resulting from all of them. This test shows how well a network trained on seven categories can select the NBV for a high score on an unseen category. We perform this procedure for *PCNBV* and *PCNBV-sph* trained in coverage and VIG scores. The results are seen in Fig. 12.



(a) Average AUC for 8-fold average validation on NBV networks

(b) Average VIG for 8-fold average validation on NBV networks

Figure 12: Average scores by view for 8-fold average validation Shapenet datasets

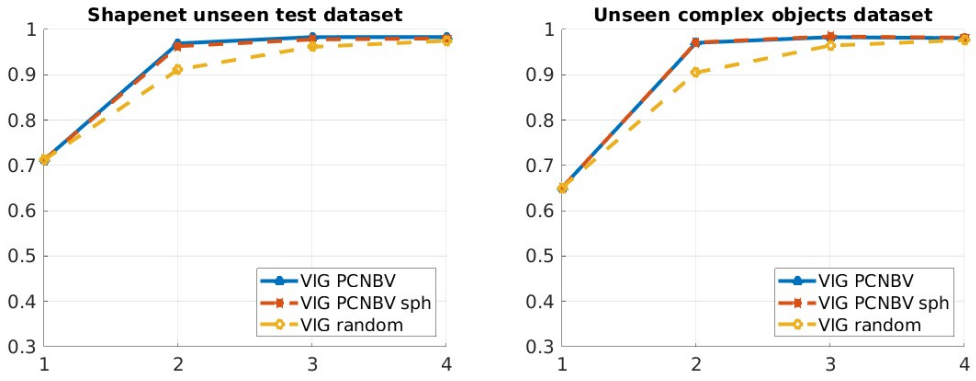
Both the coverage and VIG versions successfully use the knowledge obtained from 7 categories to estimate values for an unknown category.

We use the VIG-trained versions to estimate views that lead to a partial

point cloud with a convex hull similar to the one obtained from the densely sampled ground truth points.

5.2. VIG selection network

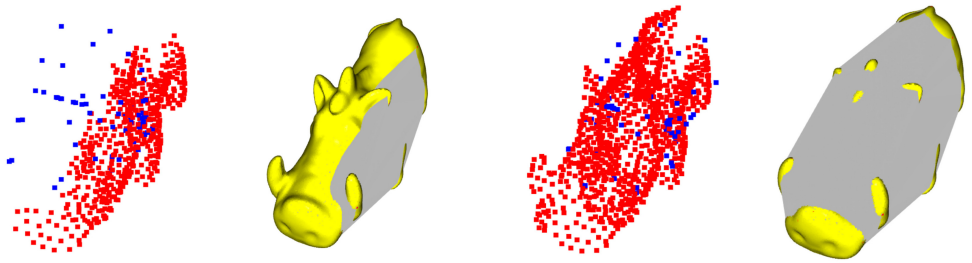
The volumetric score converges to the ground truth score with fewer views and points, as explained in the theoretical section. The average scores by views are shown in Fig. 13. *PCNBV* and *PCNBV-sph* obtain similar performance, outperforming random selection. The resulting convex hull does not reconstruct the coverage score as two views are insufficient for complete surface coverage. The convex hull encodes volumetric information from the object and can be used as a proxy geometry to predict the coverage views. A visualization of the spherical harmonics reconstruction P_{sig} for a partial point cloud P_{par} is seen in Fig. 14. The convex hull is computed for each partial point cloud P_{par} and contrasted to the original geometry.



(a) Average VIG by view on unseen Shapenet test dataset

(b) Average VIG by view on unseen complex shape dataset

Figure 13: Average VIG by view for Shapenet and complex object datasets



(a) Partial point cloud 1 and spherical signature

(b) Convex hull for Partial point cloud 1

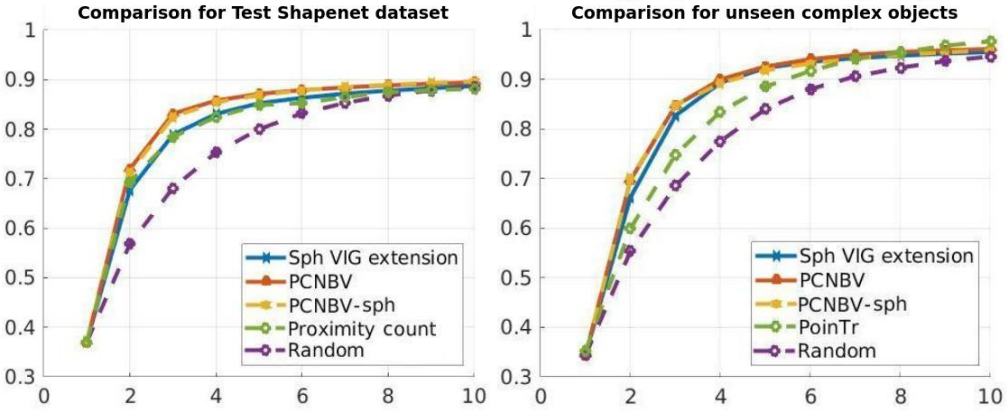
(c) Partial point cloud after prediction and spherical signature

(d) Convex hull for partial point cloud after selected view

Figure 14: Example of VIG selection with spherical harmonics signature

5.3. Coverage reconstruction comparison

The results for Shapenet unseen objects are in Fig.15. The *PCNBV* network achieves similar performance as reported in [71], replicating their results.



(a) Average AUC by view on test dataset

(b) Average AUC by view on complex dataset

Figure 15: Average AUC by view on different datasets

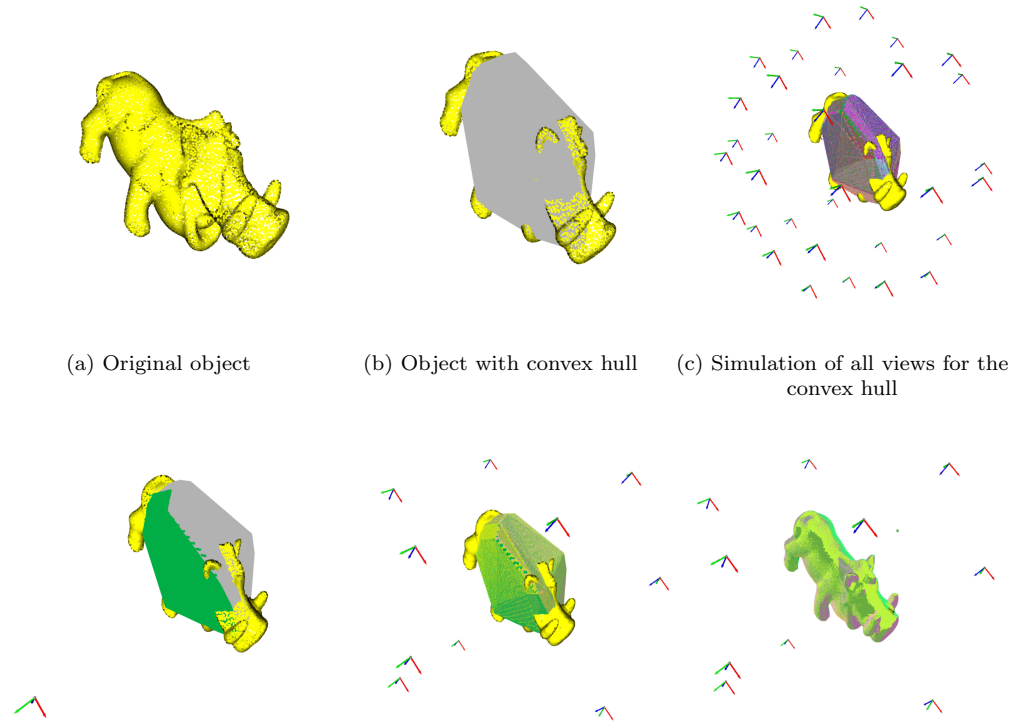
The spherical *PCNBV-sph* performs similarly to that of the *PCNBV* version. Both *PCNBV* and the set-covering methods use greedy coverage gain

and as such have similar performance but with completely different strategies. Random selection is applied on unvisited views which improves the end selection but is significantly outperformed by spherical harmonics VIG extension and the *PCNBV* network. These tests confirm that a convex hull constructed with fewer points offers sufficient information to predict the one-shot NBV. A visualization of the reconstruction process applied to a complex shape is shown in Fig. 16.

Because we compare the coverage obtained after each selected view, the one-shot selection is ordered by predicted contributions. If we are interested only in the coverage after the selection of views, then the order of the views does not matter and so path planning can be performed to minimize the camera movement.

The VIG extension has an additional advantage compared to the baseline *PCNBV* due to the ability to simulate the results for different views. Considering a mapping between the ground truth mesh and the convex hull, the predicted coverage for the convex hull will be associated with coverage obtained on the ground truth mesh. spherical harmonics VIG extension can be configured to estimate the coverage obtained after a selection of views. The complex geometry dataset is used for the remaining tests. The first test uses a view space different from that used to train the *PCNBV* network as shown in Fig. 17.

The learning methods are not able to adapt to the new circumstances and the predictions are corrupted. The convex hull methods can adapt to



(a) Original object

(b) Object with convex hull

(c) Simulation of all views for the convex hull

(d) Starting view imposed on the set covering

(e) Selected camera positions after SCO

(f) Resulting point cloud after selected views

Figure 16: Example of reconstruction using a convex hull

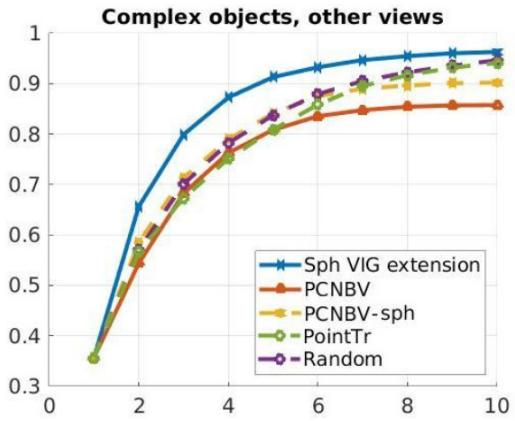


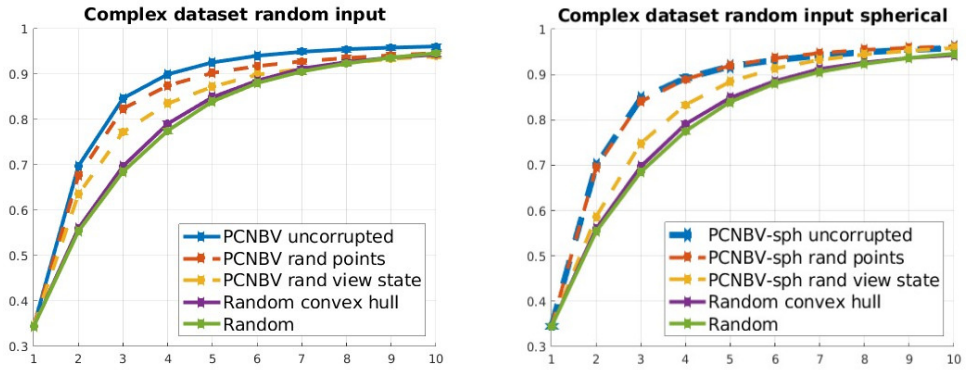
Figure 17: Average AUC by view on different view space dataset

the new views and maintain a large average AUC. The convex hull from the spherical harmonics point cloud has slightly lower performance because the shape of the signature differs more from the ground truth mesh.

5.4. Network prediction validation

To verify the validity of the NBV selection methods, we check the reconstruction process for random input. For learning methods, the input point clouds are random, or the viewstate is permuted so that the number of views is the same but the already selected views are not. We tested for changes in the geometry for each method. In the case of *PCNBV*, the sampled P_{par} which is fed into the neural network is disturbed. For spherical harmonic VIG extension, the convex hull is random. For safety checks, we replaced the point clouds with random noise to see whether the partial point cloud is used in *PCNBV*. The results of this check are shown in Fig. 18.

The results show that *PCNBV* and *PCNBV-sph* are strongly linked to



(a) Prediction check for random input on *PCNBV*

(b) Prediction check for random input on *PCNBV* with spherical harmonics

Figure 18: Disturbance rejection checks for NBV prediction methods

the view state and use it more than the input point cloud to estimate the NBV. With $l_{max} = 3$, the network is forced in training to use the viewstate, and the influence of the point cloud is not used. Permuting the viewstate has a much larger effect than offering a random point cloud but keeping the viewstate unperturbed.

5.5. Computational cost comparison

The *PCNBV* method and spherical harmonics VIG extension have differing requirements for input thus they have different computational costs. spherical harmonics VIG extension requires the convex hull and the view simulation in Blender before performing the SCO and it does not need the partial point clouds. *PCNBV* needs only the partial point clouds and crucially it needs to perform an FPS to reduce the number of points to an acceptable size. This means that the primary comparison is between the view simulation and SCO for spherical harmonics VIG extension versus the

FPS applied on large point clouds and the neural network performance. The comparison is seen in Table 1.

Method \ Model	PCNBV	PCNBV-sph	Our
FPS	3.12 s	–	–
Sph reconstruction	–	2.53 s	–
Network	0.21 s	0.02 s	–
View Simulation	–	–	3.32 s
Set Covering	–	–	0.12 s
Total	3.24 s	**2.55 s**	3.32 s
Total Online	3.24 s	2.55 s	**0.12 s**

Table 1: Time requirements for operations applied to 9-step reconstruction

An important observation is that for spherical harmonics VIG extension, the simulation of the views using the convex hull is the computationally expensive part. Still, it is not required for each reconstruction. If the convex hull and the views are available then for any starting view the SCO is performed without any need for other operations. Due to the large number of points in the resulting P_{par}, any learning method must convert P_{par} into an input with fewer points and spherical harmonics VIG extension does not need to apply this transformation.

6. Conclusions

We presented a spherical harmonics preprocessing that can speed up learning NBV methods and encode geometric priors which can be used to predict the NBV for coverage reconstruction. The proposed VIG score leads

to a convex hull using much fewer points and views. We discovered a useful link between the convex hull and the surface coverage prediction as exemplified in our VIG extension for surface coverage in which the convex hull allows a one-shot prediction of the needed views. We demonstrated that for a 3D geometry, a spherical harmonics reconstruction of a sparse point cloud can also be used to compute a convex hull that has geometric priors from the original shape. Thus, a novel connection between the VIG score, spherical harmonics reconstruction, and the convex hull has been established. Future plans include comparing the convex hull with a predicted mesh obtained from the partial point cloud, similar to [26]. Another approach of interest is to extend the NBV to collections of objects. Additionally, our method could be extended to a scene exploration context, where the first views are used to determine geometric priors which are then used to simulate the views from any camera position.

7. Credit

Alexandru Pop: Software, Data curation, Writing- Original draft preparation, Visualization, Investigation. Levente Tamas: Supervision, Methodology, Writing- Reviewing and Editing.

References

[1] Adjigble, M., Tamadazte, B., De Farias, C., Stolkin, R., Marturi, N., 2023. 3D spectral domain registration-based visual servoing, in: 2023

IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 769–775.

[2] Althloothi, S., Mahoor, M.H., Voyles, R.M., 2013. A robust method for rotation estimation using spherical harmonics representation. IEEE Transactions on Image Processing 22, 2306–2316.

[3] Alvarez, J.R., Arroqui, M., Mangudo, P., Toloza, J., Jatip, D., Rodríguez, J.M., Teyseyre, A., Sanz, C., Zunino, A., Machado, C., et al., 2018. Body condition estimation on cows from depth images using convolutional neural networks. Computers and Electronics in Agriculture 155, 12–22.

[4] Blaga, A., Militaru, C., Mezei, A.D., Tamas, L., 2021. Augmented reality integration into MES for connected workers. Robotics and Computer-Integrated Manufacturing 68, 102057.

[5] Burdziakowski, P., Tysiac, P., 2019. Combined close range photogrammetry and terrestrial laser scanning for ship hull modelling. Geosciences 9, 242.

[6] Cao, F., Shi, J., Wen, C., 2023. A dynamic graph aggregation framework for 3D point cloud registration. Engineering Applications of Artificial Intelligence 120, 105817.

[7] Chang, Y., Jung, C., Xu, Y., 2021. FinerPCN: High fidelity point cloud

completion network using pointwise convolution. Neurocomputing 460, 266–276.

[8] Chen, C., Liu, D., Xu, C., Truong, T.K., 2021. GeneCGAN: A conditional generative adversarial network based on genetic tree for point cloud reconstruction. Neurocomputing 462, 46–58.

[9] Chen, J., Zhu, F., Han, Y., Ren, D., 2023. Deep learning framework-based 3D shape reconstruction of tanks from a single RGB image. Engineering Applications of Artificial Intelligence 123, 106366.

[10] Chen, X., Chen, Y., Najjaran, H., 2020. End-to-end 3D object model retrieval by projecting the point cloud onto a unique discriminating 2D view. Neurocomputing 402, 336–345.

[11] Chen, X., Li, Q., Wang, T., Xue, T., Pang, J., 2024. GenNBV: Generalizable Next-Best-View Policy for Active 3D Reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16436–16445.

[12] Curless, B., Levoy, M., 1996. A volumetric method for building complex models from range images, in: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 303–312.

[13] Delmerico, J., Isler, S., Sabzevari, R., Scaramuzza, D., 2018. A comparison of volumetric information gain metrics for active 3D object reconstruction. Autonomous Robots 42, 197–208.

[14] Dhami, H., Sharma, V.D., Tokekar, P., 2023a. MAP-NBV: Multi-agent Prediction-guided Next-Best-View Planning for Active 3D Object Reconstruction. arXiv preprint arXiv:2307.04004 .

[15] Dhami, H., Sharma, V.D., Tokekar, P., 2023b. Pred-NBV: Prediction-guided next-best-view planning for 3D object reconstruction, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 7149–7154.

[16] Fan, H., Qi, L., Dong, J., Li, G., Yu, H., 2018. Dynamic 3D surface reconstruction using a hand-held camera, in: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, IEEE. pp. 3244–3249.

[17] Fang, Q., Song, Y., Li, K., Shen, L., Wu, H., Xiong, G., Bo, L., 2024. Evaluate geometry of radiance fields with low-frequency color prior, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1707–1715.

[18] Frohlich, R., Tamas, L., Kato, Z., 2019. Absolute pose estimation of central cameras using planar regions. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 377–391.

[19] Grieb, J., Barbero-García, I., Lerma, J.L., 2022. Spherical harmonics to quantify cranial asymmetry in deformational plagiocephaly. Scientific Reports 12, 167.

[20] Guédon, A., Monasse, P., Lepetit, V., 2022. Scone: Surface coverage optimization in unknown environments by volumetric integration. Advances in Neural Information Processing Systems 35, 20731–20743.

[21] Guédon, A., Monnier, T., Monasse, P., Lepetit, V., 2023. MACARONS: Mapping And Coverage Anticipation with RGB Online Self-Supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 940–951.

[22] Han, Y., Zhan, I.H., Zhao, W., Liu, Y.J., 2022. A double branch next-best-view network and novel robot system for active object reconstruction, in: 2022 International Conference on Robotics and Automation (ICRA), IEEE. pp. 7306–7312.

[23] Hepp, B., Dey, D., Sinha, S.N., Kapoor, A., Joshi, N., Hilliges, O., 2018. Learn-to-score: Efficient 3D scene exploration by predicting view utility, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 437–452.

[24] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N., 2013. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes, in: Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11, Springer. pp. 548–562.

[25] Hou, J., Yu, L., Fei, S., 2020. A highly robust automatic 3D reconstruction system based on integrated optimization by point line features. Engineering Applications of Artificial Intelligence 95, 103879.

[26] Hu, H., Pan, S., Jin, L., Popović, M., Bennewitz, M., 2023a. Active implicit reconstruction using one-shot view planning. arXiv preprint arXiv:2310.00685 .

[27] Hu, K., Wang, T., Shen, C., Weng, C., Zhou, F., Xia, M., Weng, L., 2023b. Overview of Underwater 3D Reconstruction Technology Based on Optical Images. Journal of Marine Science and Engineering 11, 949.

[28] Isler, S., Sabzevari, R., Delmerico, J., Scaramuzza, D., 2016. An information gain formulation for active volumetric 3D reconstruction, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 3477–3484.

[29] Jin, L., Chen, X., Rückin, J., Popović, M., 2023. NEU-nbv: Next best view planning using uncertainty estimation in image-based neural rendering, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 11305–11312.

[30] Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S., 2019. HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0.

[31] Li, J., Guo, S., Meng, X., Lai, Z., Han, S., 2022a. DPG-Net: Densely progressive-growing network for point cloud completion. Neurocomputing 491, 1–13.

[32] Li, J., Guo, S., Wang, L., Han, S., 2024. CompleteDT: Point cloud completion with information-perception transformers. Neurocomputing 592, 127790.

[33] Li, M., Huang, B., Tian, G., 2022b. A comprehensive survey on 3D face recognition methods. Engineering Applications of Artificial Intelligence 110, 104669.

[34] Li, R., Gao, R., Suganthan, P.N., Cui, J., Sourina, O., Wang, L., 2023. A spectral-ensemble deep random vector functional link network for passive brain–computer interface. Expert Systems with Applications 227, 120279.

[35] Li, Y., Qin, X., Zhang, Z., Dong, H., 2022c. Solid waste shape description and generation based on spherical harmonics and probability density function. Waste Management & Research 40, 66–78.

[36] Lin, F., Xu, Y., Zhang, Z., Gao, C., Yamada, K.D., 2022. Cosmos propagation network: Deep learning model for point cloud completion. Neurocomputing 507, 221–234.

[37] Liu, R., Li, C., Wan, W., Pan, J., Harada, K., 2024. NBV/NBC Plan-

ning Considering Confidence Obtained from Shape Completion Learning. IEEE Robotics and Automation Letters .

[38] Mendoza, M., Vasquez-Gomez, J.I., Taud, H., Sucar, L.E., Reta, C., 2020. Supervised learning of the next-best-view for 3D object reconstruction. Pattern Recognition Letters 133, 224–231.

[39] Mukhaimar, A., Tennakoon, R., Lai, C.Y., Hoseinnezhad, R., Bab-Hadiashar, A., 2022. Robust object classification approach using spherical harmonics. IEEE Access 10, 21541–21553.

[40] Naderi, H., Noorbakhsh, K., Etemadi, A., Kasaei, S., 2023. LPF-Defense: 3D adversarial defense based on frequency analysis. Plos one 18, e0271388.

[41] Pan, S., Hu, H., Wei, H., 2022. SCVP: Learning one-shot view planning via set covering for unknown object reconstruction. IEEE Robotics and Automation Letters 7, 1463–1470.

[42] Pan, S., Hu, H., Wei, H., Dengler, N., Zaenker, T., Dawood, M., Bennewitz, M., 2023. Integrating one-shot view planning with a single next-best view via long-tail multiview sampling. arXiv preprint arXiv:2304.00910 .

[43] Pan, S., Jin, L., Huang, X., Stachniss, C., Popović, M., Bennewitz, M., 2024. Exploiting Priors from 3D Diffusion Models for RGB-Based One-Shot View Planning. arXiv preprint arXiv:2403.16803 .

[44] Park, K.B., Choi, S.H., Kim, M., Lee, J.Y., 2020. Deep learning-based mobile augmented reality for task assistance using 3D spatial mapping and snapshot-based RGB-D data. Computers & Industrial Engineering 146, 106585.

[45] Patrick, S.D., Bakolas, E., 2022. Using spherical harmonics for navigating in dynamic and uncertain environments. IFAC-PapersOnLine 55, 567–572.

[46] Peng, B., Wang, W., Dong, J., Tan, T., 2021. Learning pose-invariant 3D object reconstruction from single-view images. Neurocomputing 423, 407–418.

[47] Peralta, D., Casimiro, J., Nilles, A.M., Aguilar, J.A., Atienza, R., Cajote, R., 2020. Next-best view policy for 3D reconstruction, in: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer. pp. 558–573.

[48] Pop, A., Domșa, V., Tamas, L., 2023. Rotation Invariant Graph Neural Network for 3D Point Clouds. Remote Sensing 15, 1437.

[49] Pop, A., Tamas, L., 2022. Next best view estimation for volumetric information gain. IFAC-PapersOnLine 55, 160–165.

[50] Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660.

[51] Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems 30.

[52] Radvilaitė, U., Ramírez-Gómez, Á., Kačianauskas, R., 2016. Determining the shape of agricultural materials using spherical harmonics. Computers and Electronics in Agriculture 128, 160–171.

[53] Ramasinghe, S., Khan, S., Barnes, N., Gould, S., 2020. Spectral-GANs for high-resolution 3D point-cloud generation, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 8169–8176.

[54] Slavík, P., 1996. A tight analysis of the greedy algorithm for set cover, in: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 435–441.

[55] Sun, X., Sun, S., 2021. Adversarial robustness and attacks for multi-view deep models. Engineering Applications of Artificial Intelligence 97, 104085.

[56] Tamas, L., Frohlich, R., Kato, Z., 2015. Relative pose estimation and fusion of omnidirectional and lidar cameras, in: Computer Vision-ECCV

2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II 13, Springer. pp. 640–651.

[57] Tamas, L., Jensen, B., 2014. All-season 3D object recognition challenges, in: ICRA Workshop on Visual Place Recognition in Changing Environments.

[58] Tamas, L., Murar, M., 2019. Smart CPS: vertical integration overview and user story with a cobot. International Journal of Computer Integrated Manufacturing 32, 504–521.

[59] Tang, Z., Meng, H.Y., Manocha, D., 2021. Learning acoustic scattering fields for dynamic interactive sound propagation, in: 2021 IEEE Virtual Reality and 3D User Interfaces (VR), IEEE. pp. 835–844.

[60] Wang, C., Reza, M.A., Vats, V., Ju, Y., Thakurdesai, N., Wang, Y., Crandall, D.J., Jung, S.h., Seo, J., 2024a. Deep learning-based 3D reconstruction from multiple images: A survey. Neurocomputing 597, 128018.

[61] Wang, T., Xi, W., Cheng, Y., Han, H., Yang, Y., 2024b. RL-NBV: A deep reinforcement learning based next-best-view method for unknown object reconstruction. Pattern Recognition Letters .

[62] Wang, X., Cai, M., Sohel, F., Sang, N., Chang, Z., 2021a. Adversarial point cloud perturbations against 3D object detection in autonomous driving systems. Neurocomputing 466, 27–36.

[63] Wang, X., Yin, Z.y., Zhang, J.q., Xiong, H., Su, D., 2021b. Three-dimensional reconstruction of realistic stone-based materials with controllable stone inclusion geometries. Construction and Building Materials 305, 124240.

[64] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph CNN for learning on point clouds. ACM Transactions On Graphics (TOG) 38, 1–12.

[65] Wen, Y., Lin, J., Chen, K., Chen, C.P., Jia, K., 2020. Geometry-aware generation of adversarial point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 2984–2999.

[66] Wu, C., Zeng, R., Pan, J., Wang, C.C., Liu, Y.J., 2019. Plant phenotyping by deep-learning-based planner for multi-robots. IEEE Robotics and Automation Letters 4, 3113–3120.

[67] Wu, J., Wyman, O., Tang, Y., Pasini, D., Wang, W., 2024. Multi-view 3D reconstruction based on deep learning: A survey and comparison of methods. Neurocomputing 582, 127553.

[68] Yang, Y., Chen, F., Wu, F., Zeng, D., Ji, Y.m., Jing, X.Y., 2020. Multi-view semantic learning network for point cloud based 3D object detection. Neurocomputing 397, 477–485.

[69] Yifan, W., Wu, S., Huang, H., Cohen-Or, D., Sorkine-Hornung, O., 2019. Patch-based progressive 3D point set upsampling, in: Proceed-

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5958–5967.

[70] Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J., 2021. Pointr: Diverse point cloud completion with geometry-aware transformers, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 12498–12507.

[71] Zeng, R., Zhao, W., Liu, Y.J., 2020. PC-NBV: A point cloud based deep network for efficient next best view planning, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 7050–7057.

[72] Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2019. Self-attention generative adversarial networks, in: International Conference on Machine Learning, PMLR. pp. 7354–7363.

[73] Zhang, Q., Baek, S.H., Rusinkiewicz, S., Heide, F., 2022. Differentiable point-based radiance fields for efficient view synthesis, in: SIGGRAPH Asia 2022 Conference Papers, pp. 1–12.

[74] Zhao, M., Xiong, G., Zhou, M., Shen, Z., Wang, F.Y., 2021. 3D-RVP: A method for 3D object reconstruction from a single depth view using voxel and point. Neurocomputing 430, 94–103.

[75] Zhou, Q.Y., Park, J., Koltun, V., 2018. Open3D: A modern library for 3D data processing. arXiv preprint arXiv:1801.09847 .

[76] Zhu, L., Wang, B., Tian, G., Wang, W., Li, C., 2021. Towards point cloud completion: Point rank sampling and cross-cascade graph CNN. Neurocomputing 461, 1–16.

[77] Zimny, D., Waczyńska, J., Trzciński, T., Spurek, P., 2024. Points2NERF: Generating neural radiance fields from 3D point cloud. Pattern Recognition Letters 185, 8–14.