Semantic 3D Reconstruction of a Scene and Its Effective Visualisation

Vladimir V. Kniaz[1,2], Petr V. Moshkantsev[1], Artem N. Bordodymov[1], Vladimir A. Mizginov[1] and Daniil I. Novikov[1]

[1]State Res. Institute of Aviation Systems (GosNIIAS), 125319, 7, Victorenko str., Moscow, Russia
[2]Moscow Institute of Physics and Technology (MIPT), Russia

Abstract

Single-image 3D scene reconstruction is required in multiple challenging tasks including mobile robotics, industrial monitoring and reconstruction of lost cultural heritage. While modern models demonstrate robust resolution of scene in real time with resolution up to $128 \times 128 \times 128$ voxels, visualization of such detailed of a such detailed voxel model is challenging. A model with 128^3 voxels contains 2097152 simple cubes 16M vertices. It is unfeasible for modern hardware to perform visualization of such voxel models in real-time. Hence a voxel model simplification technique is required to demonstrate reconstruction results in real-time. In this paper, we propose a new algorithm for voxel model simplification using predefined camera views. The algorithm reduces a rigid-body voxel model to a shell voxel model. It keeps only the voxels that are visible from the required view. We demonstrate the effectiveness of the proposed algorithm using a case study with a mobile robot and a state-of-the-art SSZ single-photo 3D reconstruction neural network. We generated a real and a virtual scene with various objects including a statue. We use a mobile robot equipped with a single camera to collect real and synthetic data. We train the SSZ model using the collected data. We developed a dedicated visualization software that implements our algorithm. The comparison of the visualization performance for the full model and its reduced version demonstrates that our algorithm allows to increase the performance by 420 times.

Keywords

voxel model visualization, single-photo 3D reconstruction, scientific visualization, neural networks.

1. Introduction

Single-image 3D reconstruction is required in multiple fields including mobile robotics [1], industrial monitoring and reconstruction of lost cultural heritage [2]. Modern neural networks can perform simultaneous 3D reconstruction of the scene and its semantic segmentation [3]. The result is a high-resolution voxel model with 128^3 voxels. While

GraphiCon 2022: 32nd International Conference on Computer Graphics and Vision, September 19–22, 2022,
Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russia
EMAIL: vl.kniaz@gosniias.ru (V. V. Kniaz); petr-mosh@gosniias.ru (P. V. Moshkantsev); bordodymov@gosniias.ru (A. N. Bordodymov); vl.mizginov@gosniias.ru (V. A. Mizginov); daninov@gosniias.ru (D. I. Novikov)
ORCID: 0000-0003-2912-9986 (V. V. Kniaz); 0000-0001-9624-4322 (P. V. Moshkantsev); 0000-0001-8159-2375 (A. N. Bordodymov); 0000-0003-1885-3346 (V. A. Mizginov)

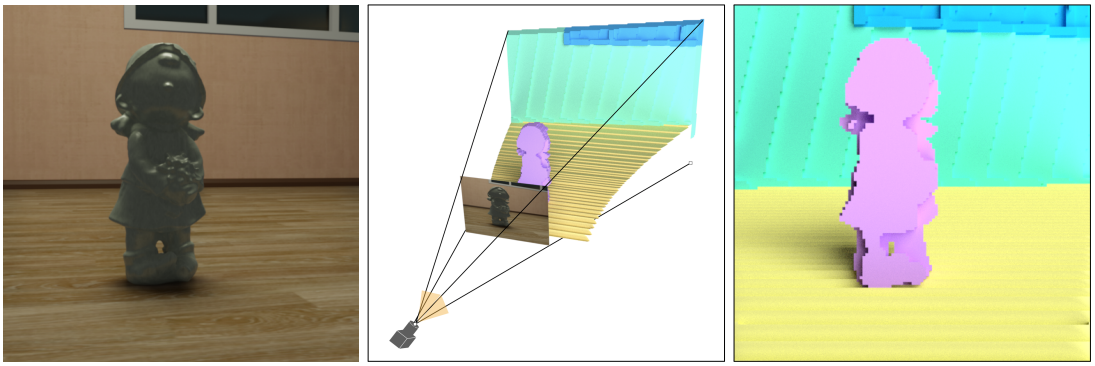

Figure 1: Example of single-image 3D-reconstruction. Single input image (left), camera frustum and the frustum voxel model (center), semantic voxel model output (right)

modern models can generate such voxel models in quasi real-time, visualization of such detailed voxel model is challenging (Figure 7).

The main complexity of this task arises from the large number of voxels that are invisible for the virtual camera during the visualization. Still, elementary cubes must be generated for each frame as the voxel model is being updated by a neural network in the real-time.

The visualization of voxel models has been intensively studied recently [4, 5, 6, 7, 8, 9, 10]. Multiple approaches has been proposed to improve the visualization performance. Still, to the best of our knowledge there has been no research to date regarding effective visualization of semantic voxel models.

This paper is focused on the development of a voxel model simplification algorithm (`VMS`). Our `VMS` algorithm aims reducing the original semantic voxel model representing a rigid body to a shell-voxel model that includes only the surfaces of objects in the scene that are visible from a given viewpoint. To achieve this, we use a preliminary ray-tracing stage that allows us to find sets of voxels that are visible from a given viewpoint. During the inference, we keep only first non-zero element in a voxel set for each pixel.

We developed an environment simulator to train and validate our `VMS` algorithm. Our virtual scene represents a room with various object of eight classes. We developed a new *RoboticVoxels* dataset with 16k samples using our environment simulator. We use a state-of-the-art single-photo 3D reconstruction model [3] to evaluate our `VMS` algorithm. The results of the evaluation are encouraging and demonstrate that our `VMS` algorithm allows to improve the rendering performance by 420 times.

2. Related work

2.1. Single-image 3D Reconstruction

Obtaining an accurate 3D model of an object from its single image as an input is quite difficult. Since the 2000s, this problem has been intensively studied [11, 12]. The development of neural network technologies has led to the emergence of new algorithms

based on deep convolutional neural network [13, 14, 15, 16, 17]. Methods have been proposed to predict unobserved voxels from a single depth map [18, 19, 20], however, prediction of a voxel model of a complex scene from a single color (RGB) image is more variable. Object-centered models [21] reconstruct object 3D model in the same coordinate system for any camera pose with respect to the object. The solution of the problem occurs in 2 steps: object recognition and a 3D shape reconstruction. The method leverages an auto-encoder architecture for a voxel model prediction. The method showed encouraging results, but the resolution of the model was only $20 \times 20 \times 20$ elements.

The most accurate results were obtained by methods based on generative adversarial neural networks [22, 23]. Methods that leverage a latent space for 3D shape synthesis were developed recently. Our paper [2] describes image-to-voxel translation network (Z-GAN) as a starting point. Z-GAN network utilizes the skip connections in the generator network to transfer 2D features to a 3D voxel model effectively. Therefore, the network can generate voxel models of previously unseen objects using object silhouettes present on the input image and the knowledge obtained during a training stage. The other our paper [3] represent a single shot image-to-semantic voxel model translation framework. We train a generator adversarially against a discriminator that verifies the object's poses. Furthermore, trapezium-shaped voxels, volumetric residual blocks, and 2D-to-3D skip connections facilitate our model learning explicit reasoning about 3D scene structure. In [24] a novel framework for single-view and multi-view 3D object reconstruction was proposed. By using a well-designed encoder-decoder, it generates a coarse 3D volume from each input image. A multi-scale context-aware fusion module is then introduced to adaptively select high-quality reconstructions for different parts from all coarse 3D volumes to obtain a fused 3D volume.

Finally, the most modern transformer neural networks are used in 3D reconstruction tasks. In [25] the multi-view 3D reconstruction is presented as a sequence-to-sequence prediction problem and is proposed a framework named 3D Volume Transformer.Unlike previous CNN-based methods using a separate design, it combines feature extraction and view fusion in one Transformer network.

2.2. Volumetric Neural Networks

Volumetric neural networks is an extension from 2D CNNs which takes hierarchical 3D image information by dividing it into small cubes, instead of taking 2D patches, to capture discriminative features along both the spatial and the temporal dimensions. This introduces challenges for learning-based approaches, as 3D object annotations are scarce in real images. Previous work chose to train on synthetic data with ground truth 3D information, but suffered from domain adaptation when tested on real data. In [26] a two-step approach for 3D shape reconstruction via 2.5D sketches was presented. This method has several advantages. The first of them is the ability to learn only on synthesized data. Second, compared to full 3D shape, 2.5D sketches are much easier to be recovered from a 2D image. Third, differentiable projective functions is it derived from 3D shape to 2.5D sketches.

Wu et al. have proposed a novel framework [27], namely 3D Generative Adversarial

Network (3D-GAN), which generates 3D objects from a probabilistic space by leveraging recent advances in volumetric convolutional networks and generative adversarial nets. The use of an adversarial criterion, instead of traditional heuristic criteria, enables the generator to capture object structure implicitly and to synthesize high-quality 3D objects. The generator establishes a mapping from a low-dimensional probabilistic space to the space of 3D objects, so that objects can be sampled without a reference image or CAD models, and explore the 3D object manifold. The adversarial discriminator provides a powerful 3D shape descriptor which, learned without supervision, has wide applications in 3D object recognition. This made it possible to predict models with a resolution $64 \times 64 \times 64$ elements from a randomly sampled noise vector. A method based on combining the architectures of volumetric and multi-view neural networks is presented in the work [28]. Volumetric neural networks are also actively used to work with three-dimensional point clouds [29, 30]. In [31] the task of searching for a query object of unknown position and pose in a scene, both given in the form of 3D point cloud data, was studied. This method includes a deep reinforcement learning approach that jointly learns both the features and the efficient search path.This network is successfully trained in an end-to-end manner by integrating a contrastive loss and a reinforcement localization reward. The important problem of accelerating the processing of three-dimensional point clouds was solved in [30]. The algorithm diverse mapping operations onto one versatile ranking-based kernel, streams the sparse computation with configurable caching, and temporally fuses consecutive dense layers to reduce the memory footprint and achieves 3.7× speedup and 22× energy savings. Volumetric neural networks are actively used to solve the problem of recognition of three-dimensional images from two-dimensional input data [32, 33, 34].

2.3. Voxel Model Visualisation

Visualization is of great interest in areas where inner substances of an object are to be studied. Computer tomography, widely exploited in medcine, produces large amounts of data that needs to be effectively visualized. So great part of studies on voxel model visualization was carried out on medical data. Form the first works, researchers tried to find techniques for fast and realistic visualization [4, 5].

The Far Voxels [6] method, that integrates visibility culling and out-of-core data management with a level-of-detail framework, allowed to improve the efficiency and generality of very large arbitrary surface models. It generates a coarse volume hierarchy by binary space partitioning at the preprocessing time. When rendering, the volumetric structure is refined and rendered in front-to-back order. The performance of such approach was demonstrated on extremely complex heterogeneous surface models. With some modifications [7], it fairly represents small or thin CAD models of hundreds of millions of triangles, which is especially visible during transitions between different levels of detail.

With the advances in machine learning methods, they were successfully applied for realistic voxel model visualization. The VoxelEmbed method [8] provides simultaneous cell instance segmenting and tracking on 3D volumetric video sequences. The VoxelRend-

based network (VR-U-Net) [9] combines a memory-efficient variant of 3D U-Net with a voxel-based rendering (VoxelRend) module that refines local details by voxel-based predictions on non-regular grids. Experimental evaluation demonstrated that the proposed VR-U-Net is memory-efficient and provides high-quality segmentation results.

Usually, the main problem in a voxel 3D model representation is a huge volume of the 3D model, that is needed for high quality visualization. The multi-level voxel representation based on linear segmentation method was proposed to find a solution for efficiently representing and the geological structures and internal non-uniform properties of tunnel engineering [10]. The method uses Volumetric Dynamic B + trees (VDB) data structure for integrating and updating models. The evaluation of such approach showed improving spatial efficiency for 28.49% after segmentation, and data access with $O(1)$ time complexity.

3. Method

3.1. Frustum Voxel Models

A fruxel 3D model is made up of elements called fruxels. They differ from rectangular voxels in that they are trapezoid-shaped. A ray passes through each fruxel, connecting a pixel on the camera matrix and a point on an object in the scene. If it is necessary to predict n classes of objects in an image, then the semantic voxel model $F \in \{0, 1, \ldots, n-1\}^{w \times h \times d}$ is a three-dimensional tensor that contains the object class number $i \in I$ in a given fruxel.

Therefore, this fruxel model is a multi-layered semantic segmentation. Each slice is the result of the intersection of the object and a plane orthogonal to the optical axis of the camera, located at a given distance. The fruxel model is described by a set of parameters $\{z_n, z_f, d, \alpha\}$, where z_n is the distance from the camera to the nearest clipping plane, z_f is the distance to the far clipping plane, d is the number of slices, and α is the horizontal field of view of the camera (see Figure 1).

3.2. SSZ Neural Network

The main difficulty in translation an image into a 3D voxel model is that it is necessary to convert high-resolution 2D objects into 3D. This transformation can be done using latent space and the use of skip joins, which improve the model's ability to generalize. Our model `SSZ` is based on the works [35, 36] , in which the corresponding convolution levels in the generator decoder and encoder are interconnected using skipped connections.

The dimension of the feature map in our model generator encoder is 3 ($\boldsymbol{N}_e \in R^{w \times h \times c}$, $w \times h$ is the slice dimension, d is number of slices), while the decoder has 4 ($\boldsymbol{N}_d \in R^{w \times h \times d \times c}$, where c is the number of channels). In order for the dimensions of the encoder and decoder layers to match, we copy d two-dimensional slices into the encoder feature maps. This operation does not add new information about objects, but pixel-based contour matching allows the model to explicitly match 2D object contours and 3D shapes.

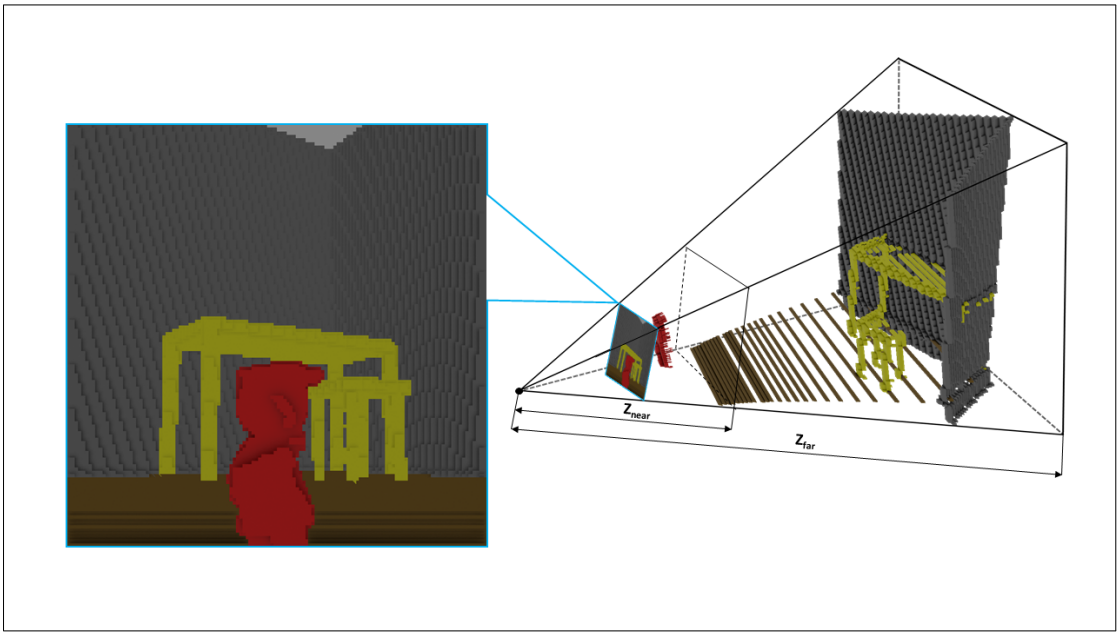

Figure 2: Slices generation by the boolean intersection of a cutting plane with 3D objects

We have designed our network architecture using inverted residual blocks [37, 38] and an additional pointwise and depthwise convolutions that downscale the feature map. This allows for more efficient propagation of the gradient. Also due to its architecture, our model works near real-time inference time.

We use volumetric inverted residual blocks to construct our decoder. Each volumetric inverted residual block includes a volumetric depth separable deconvolution layer followed by a Leaky ReLU activation and a pointwise volumetric convolution. We believe that depth separable convolution in our volumetric inverted residual blocks facilitates learning diverse filters for 2D and 3D features maps. The resulting generator architecture is presented in Figure 3.

3.3. Environment Simulator

We use the Blender 3D modeling tool as an environment simulator. A three-dimensional scene of the room was created, in which there was an object model (gnome) and a virtual camera. An examples of an image of a virtual environment is shown in the figures 4 , 5.

To form slice images, the JSON-RPC library in python was developed. Ray tracing scripts have also been developed for building fruxel models and virtual camera movement. Examples of the built fruxel scene are shown in the Figure. 6

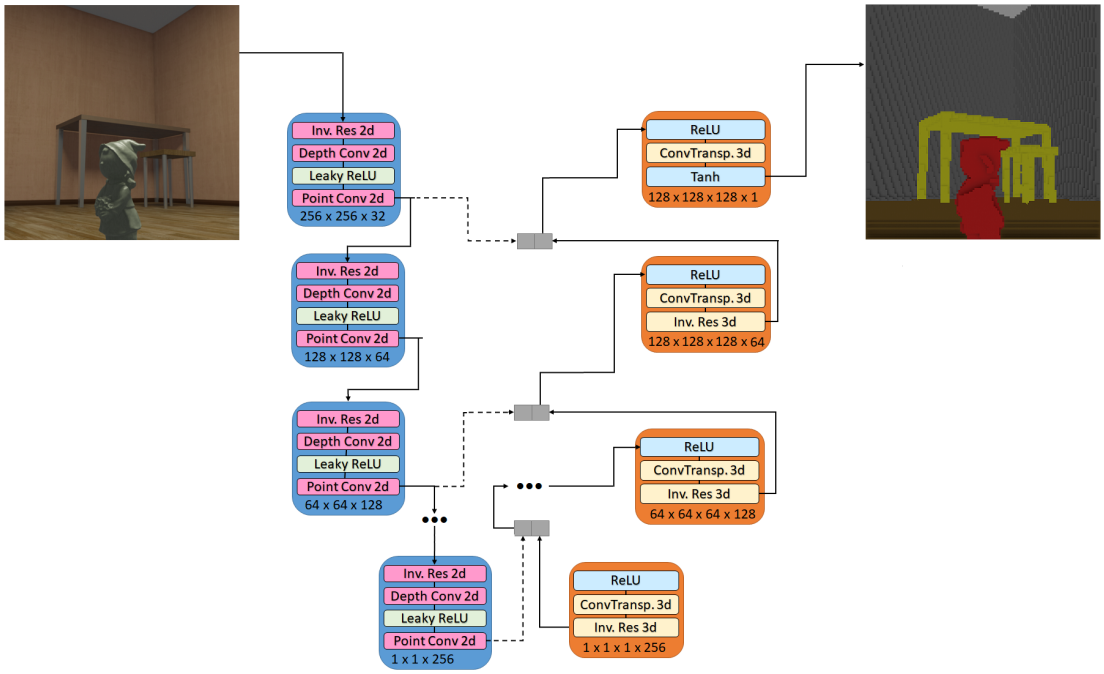

Figure 3: The generator architecture

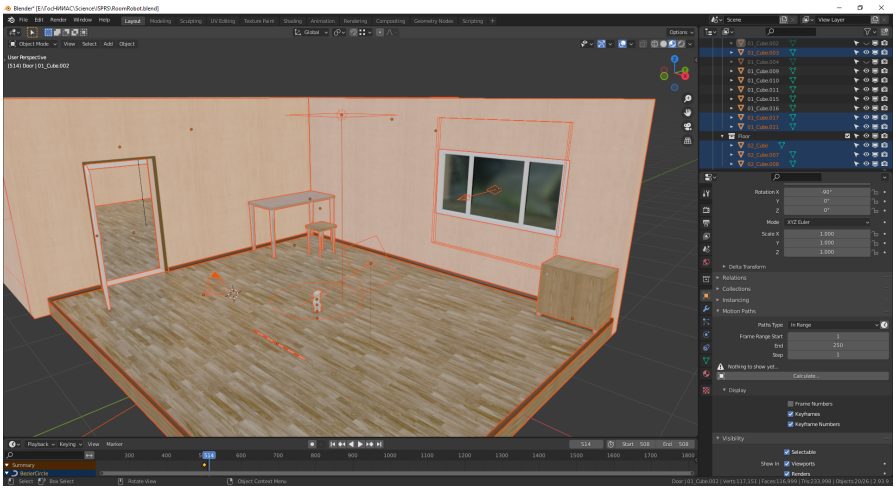

Figure 4: The general view of the environment simulator in Blender 3D

3.4. Mobile Robot

To evaluate the work of the developed algorithm, a simplified mathematical model of the movement of a four-wheeled robot in virtual space was created [39]. It is assumed that the inertia of the rotation of the wheels is negligible, and the friction that slows down the

Figure 5: Location points of the virtual camera in the scene

movement of the car is proportional to the speed with a constant of proportionality b. Then robot can be approximated for modeling purposes.In this case in inertial acceleration is simply the second derivative of x ($a = \ddot{x}$) because the robot position is measured with respect to an inertial reference frame. Since the variable of interest to us is the displacement, the equation of motion of the robot will take the form:

$$\ddot{x} + \frac{b}{m}\dot{x} = \frac{u}{m}$$

Let us introduce the notation $\frac{b}{m} = a$ and $\frac{1}{m} = b$. Then, making a change of variables, we obtain a system of differential equations:

$$x = x_1, \dot{x} = x_2$$

$$\begin{cases} \dot{x_1}{=}x_2, \\ \dot{x_2}{=}{-}ax_1 + bu, \end{cases}$$

In state-space representation, this system takes the form

$$\dot{X} = Ax + Bu,$$

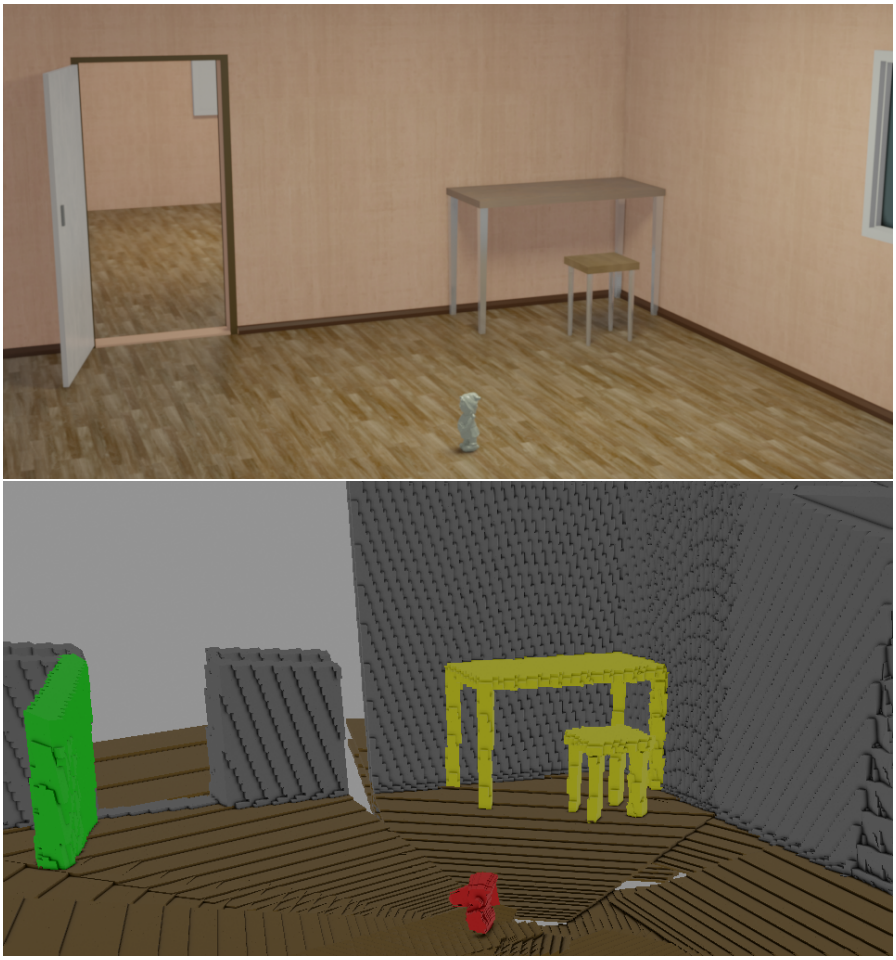

Figure 6: Examples of color images and ground truth semantic fruxel models

where $A = \begin{pmatrix} 0 & 1 \\ 0 & \text{-a} \end{pmatrix}$, $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ b \end{pmatrix}$, $u^{\mathrm{T}} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$.

The components of the state vector of the system x is the value of the longitudinal displacement (x_1) and longitudinal velocity (x_2). The control action u is a vector $(u_r; u_l)$. Its components are the speed of rotation of the left and right wheels, which are limited to a range of values $[-100; 100]$. The initial position in the Cartesian coordinate system on the OXY plane is given by the variables x_0 and y_0. Initial speed and initial heading angle are v_0 and θ_0. All initial conditions are equal to zero.

Motion simulation is performed with a given time step δt. At each step, displacement and speed values are calculated. Turning the wheeled robot is done by changing the heading angle θ. The value and direction of this angle depend on the difference in the speed of rotation of the left and right wheels $\theta(t+1) = \theta(t) + \frac{(u_r - u_l)}{50} \cdot 2\pi \cdot \delta t$. Thus, at each moment of time, the current position of the mobile robot is calculated, as well as the values of the angle of rotation and its direction.

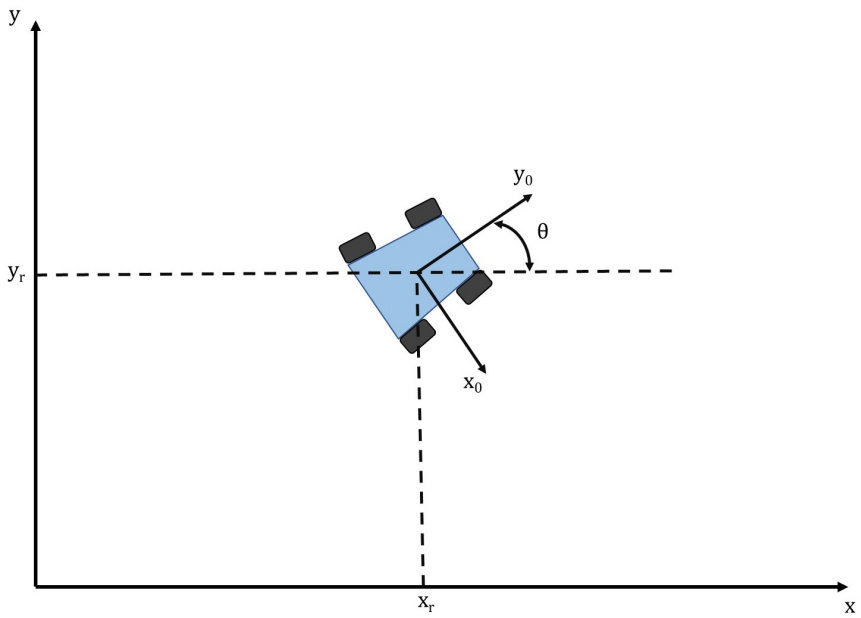

Figure 7: The scheme of the movement of the robot. Yaw angle θ shows the deviation of the direction of movement of the robot from the longitudinal axis

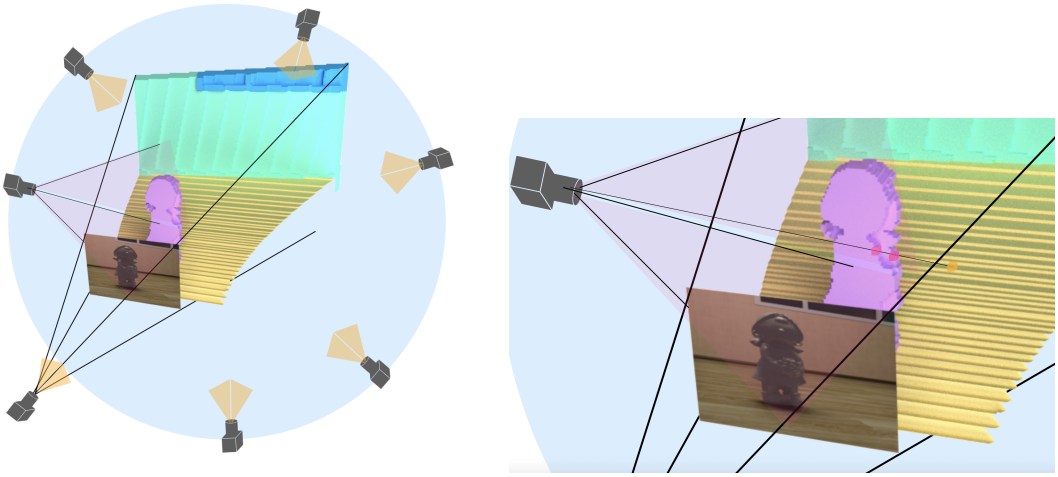

Figure 8: Voxel model simplification algorithm: predefined camera poses (left), estimation of the sets of visible voxels using ray-tracing.

3.5. Voxel Model Simplification Algorithm

Our proposed voxel model simplification algorithm (`VMS`) aims converting a rigid body semantic voxel model to a shell-voxel model that includes only such voxels that belong to the visible surfaces of objects in the scene. Let $F \in \{0, 1, \ldots, n-1\}^{w \times h \times d}$ be the input semantic voxel model. Let the 'empty' space be encoded with the class label $l = 0$. Then,

Table 1
FPS for various visualization approaches

Visualization approach	FPS
No simplification	1/30
VMS	14

our aim is designing such algorithm that provides a mapping $V : \{F, p\} \rightarrow F'$, where F' is a simplified semantic voxel model in which each voxel element is equal to 0 if it is invisible from a given camera pose, and p is the camera pose.

Our approach is straightforward. We define m possible camera poses with respect to the semantic voxel model that must be visualized (Figure 8, left). For each camera pose, we perform ray tracing from the optical center of the camera through each pixel to find a set of voxels $W = \Big\{\{x_1, y_1, z_1\}, \{x_2, y_2, z_2\}, \dots\Big\}$ that lie on a given ray (Figure 8, right). The ray-tracing is performed only once and the generated indices are stored in the cache.

After that during the inference, we find the closest predefined pose for a given input pose p. We have a precomputed set of possibly visible voxels $W(x, y)$ for each image pixel (x, y). For each pixel, we find the first non-zero voxel in the set $W(x, y)$. We keep the class value of this voxel and set all other voxels equal to zero. To perform visualization, we convert each frustum voxel to a trapezium that occupies the given volume.

4. Evaluation

4.1. Dataset Generation

We use our environment simulator to generate out *RoboticVoxels* dataset. The dataset includes 16k samples consisting of pairs of images and the corresponding semantic voxel models. The images present the virtual scene with objects of six classes: wall, floor, window, furniture, door, sculpture. The resolution of color images is 512 by 512 pixels. The resolution of semantic voxel models is $128 \times 128 \times 128$. The dataset is split into a training set with 15k samples and a test set with 1k samples. We use various augmentation techniques to increase the dataset diversity. Example images from the dataset are presented in Figure 9.

4.2. Quantitative Evaluation

We perform the quantitative evaluation of our VMS algorithm in terms of the possible maximum frames per second (FPS) for visalization of a given semantic voxel model. We compare our VMS algorithm to a straightforward visualization of non-simplified voxel model. Our approach allows to improve the performance by 420 times and achieve a quasi-real-time FPS of 14 frames per second.

The results of performance evaluation is shown in Table 1.

Figure 9: Examples of color images from our *RoboticVoxels* dataset

5. Conclusion

We proposed a new algorithm for voxel model simplification using predefined camera views. The algorithm reduces a rigid-body voxel model to a shell voxel model. It keeps only the voxels that are visible from the required view. We demonstrate the effectiveness of the proposed algorithm using a case study with a mobile robot and a state-of-the-art `SSZ` single-photo 3D reconstruction neural network. We generated a real and a virtual scene with various objects including a statue. We use a mobile robot equipped with a single camera to collect real and synthetic data. We train the `SSZ` model using the collected data. We developed a dedicated visualization software that implements our algorithm. The comparison of the visualization performance for the full model and its reduced version demonstrates that our algorithm allows to increase the performance by 420 times.

Acknowledgments

The reported study was supported by National Centre for Physics and Mathematics according to the research project 9 "Artificial Intellect and Big Data in Technical, Industrial, Environmental and Social Systems" (NCFM-9-GosNIIAS).

References

[1] V. Knyaz, V. Kniaz, Object recognition for UAV navigation in complex environment, in: L. Bruzzone, F. Bovolo, E. Santi (Eds.), Image and Signal Processing for Remote Sensing XXVI, volume 11533, International Society for Optics and Photonics, SPIE, 2020, p. 115330P. URL: https://doi.org/10.1117/12.2574078. doi:10.1117/12.2574078.

[2] V. V. Kniaz, F. Remondino, V. A. Knyaz, Generative adversarial networks for single photo 3d reconstruction, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W9 (2019) 403–408. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W9/403/2019/. doi:10.5194/isprs-archives-XLII-2-W9-403-2019.

[3] V. V. Kniaz, V. A. Knyaz, F. Remondino, A. Bordodymov, P. Moshkantsev, Image-to-voxel model translation for 3d scene reconstruction and segmentation, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 105–124.

[4] R. Kitney, L. Moura, K. Straughan, 3-d visualization of arterial structures using ultrasound and voxel modelling, Int J Cardiac Imag 4 (1989) 135–143. URL: https://doi.org/10.1007/BF01745143.

[5] K. Höhne, M. Bomans, A. Pommert, M. Riemer, C. Schiers, U. Tiede, G. Wiebecke, 3d visualization of tomographic volume data using the generalized voxel model, The Visual Computer 6 (1990) 28–36. URL: https://doi.org/10.1007/BF01902627.

[6] E. Gobbetti, F. Marton, Far voxels: A multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms, ACM Trans. Graph. 24 (2005) 878–885. URL: https://doi.org/10.1145/1073204.1073277. doi:10.1145/1073204.1073277.

[7] G. N. Wagner, A. Raposo, M. Gattass, An anti-aliasing technique for voxel-based massive model visualization strategies, in: G. Bebis, R. Boyle, B. Parvin, D. Koracin, N. Paragios, S.-M. Tanveer, T. Ju, Z. Liu, S. Coquillart, C. Cruz-Neira, T. Müller, T. Malzbender (Eds.), Advances in Visual Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 288–297. doi:10.1007/978-3-540-76858-6{_}29.

[8] M. Zhao, Q. Liu, A. Jha, R. Deng, T. Yao, A. Mahadevan-Jansen, M. J. Tyska, B. A. Millis, Y. Huo, Voxelembed: 3d instance segmentation and tracking with voxel embedding based deep learning, in: C. Lian, X. Cao, I. Rekik, X. Xu, P. Yan (Eds.), Machine Learning in Medical Imaging, Springer International Publishing, Cham, 2021, pp. 437–446.

[9] Q. Liu, C. Lian, D. Xiao, L. Ma, H. Deng, X. Chen, D. Shen, P.-T. Yap, J. J. Xia, Skull segmentation from cbct images via voxel-based rendering, in: C. Lian, X. Cao, I. Rekik, X. Xu, P. Yan (Eds.), Machine Learning in Medical Imaging, Springer International Publishing, Cham, 2021, pp. 615–623. doi:10.1007/978-3-030-87589-3{_}63.

[10] H. Wu, Q. Zhu, Y. Guo, W. Zheng, L. Zhang, Q. Wang, R. Zhou, Y. Ding, W. Wang, S. Pirasteh, M. Liu, Multi-level voxel representations for digital twin models of tunnel geological environment, International Journal of Applied Earth Observation

and Geoinformation 112 (2022) 102887. URL: https://www.sciencedirect.com/science/article/pii/S1569843222000899. doi:https://doi.org/10.1016/j.jag.2022.102887.

[11] F. Remondino, A. Roditakis, Human figure reconstruction and modeling from single image or monocular video sequence, in: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings., 2003, pp. 116–123. doi:10.1109/IM.2003.1240240.

[12] S. El-Hakim, A flexible approach to 3d reconstruction from single images, in: Acm Siggraph, volume 1, 2001, pp. 12–17.

[13] C. B. Choy, D. Xu, J. Gwak, K. Chen, S. Savarese, 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016.

[14] R. Girdhar, D. F. Fouhey, M. Rodriguez, A. Gupta, Learning a predictable and generative vector representation for objects, in: European Conference on Computer Vision, Springer, 2016, pp. 484–499.

[15] Q. Huang, H. Wang, V. Koltun, Single-view reconstruction via joint analysis of image and shape collections, ACM Transactions on Graphics 34 (2015) 87:1–87:10.

[16] D. Shin, C. Fowlkes, D. Hoiem, Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[17] D. Shin, Z. Ren, E. B. Sudderth, C. C. Fowlkes, 3d scene reconstruction with multi-layer depth and epipolar transformers, in: The IEEE International Conference on Computer Vision (ICCV), 2019.

[18] M. Firman, O. Mac Aodha, S. Julier, G. J. Brostow, Structured prediction of unobserved voxels from a single depth image, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[19] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, T. Funkhouser, Semantic scene completion from a single depth image, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[20] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, N. Trigoni, 3d object reconstruction from a single depth view with adversarial learning, in: The IEEE International Conference on Computer Vision (ICCV) Workshops, 2017.

[21] R. Girdhar, D. F. Fouhey, M. R. E. C. on, 2016, Learning a predictable and generative vector representation for objects, Springer (????) 702–722.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[23] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5967–5976.

[24] H. Xie, H. Yao, X. Sun, S. Zhou, S. Zhang, Pix2vox: Context-aware 3d reconstruction from single and multi-view images, in: The IEEE International Conference on Computer Vision (ICCV), 2019.

[25] D. Wang, X. Cui, X. Chen, Z. Zou, T. Shi, S. E. Salcudean, Z. J. Wang, R. K. Ward,

Multi-view 3d reconstruction with transformers, 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 5702–5711.

[26] J. Wu, Y. Wang, T. Xue, X. Sun, W. T. Freeman, J. B. Tenenbaum, MarrNet: 3D Shape Reconstruction via 2.5D Sketches, in: Advances In Neural Information Processing Systems, 2017.

[27] J. Wu, C. Zhang, T. Xue, W. T. Freeman, J. B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, in: Advances in Neural Information Processing Systems, 2016, pp. 82–90.

[28] C. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L. J. Guibas, Volumetric and multi-view cnns for object classification on 3d data, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 5648–5656.

[29] Y. Lin, Z. Zhang, H. Tang, H. Wang, S. Han, Pointacc: Efficient point cloud accelerator, MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture (2021).

[30] A. Goyal, H. Law, B. Liu, A. Newell, J. Deng, Revisiting point cloud shape classification with a simple and effective baseline, in: ICML, 2021.

[31] O. Krishna, G. Irie, X. Wu, T. Kawanishi, K. Kashino, Adaptive spotting: Deep reinforcement object search in 3d point clouds, in: ACCV, 2020.

[32] J. Xu, X. Zhang, W. Li, X. Liu, J. Han, Joint multi-view 2d convolutional neural networks for 3d object classification, in: IJCAI, 2020.

[33] X. Wei, R. Yu, J. Sun, View-gcn: View-based graph convolutional network for 3d shape analysis, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1847–1856.

[34] S. A. Khan, Y. Shi, M. Shahzad, X. Zhu, Fgcn: Deep feature-based graph convolutional network for semantic segmentation of urban 3d point clouds, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2020) 778–787.

[35] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks (2018) 4510–4520.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR abs/1512.03385 (2015). URL: http://arxiv.org/abs/1512.03385. arXiv:1512.03385.

[38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, L. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 4510–4520. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html. doi:10.1109/CVPR.2018.00474.

[39] V. V. Kniaz, Fast instantaneous center of rotation estimation algorithm for a skied-steered robot, in: F. Remondino, M. R. Shortis (Eds.), Videometrics, Range Imaging, and Applications XIII, volume 9528, International Society for Optics and Photonics, SPIE, 2015, pp. 194 – 204. doi:10.1117/12.2184834.