



Neural rendering-enabled 3D modeling for rapid digitization of in-service products

Jianjing Zhang^a, Sichao Liu^b, Robert X. Gao (1)^{a,*}, Lihui Wang (1)^b

^a Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA

^b Department of Production Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:

Available online 24 May 2023

Keywords:

Digital twin
Neural network
3D modeling

ABSTRACT

Rapid digitization of physical objects enables monitoring, analysis, and maintenance of in-service products, of which an up-to-date CAD model is not available. It provides designers with the products' actual response to the real-world usage, which provides a reference base for design optimization. This paper presents neural rendering as a novel method for rapid digital model building. It learns a radiance field from RGB images to determine the characteristics of the physical object. Textured mesh can be generated from the learned radiance field for efficient 3D modeling. The effectiveness of the method is demonstrated by an engine component.

© 2023 CIRP. Published by Elsevier Ltd. All rights reserved.

1. Introduction

As a product enters its service life, usage-induced degradation can cause deviation of the product from its initial CAD model, due to the effects of mechanical, thermal, chemical, or other environmental conditions over the period of its usage. The capability of digitizing a product at any stage during its life cycle not only enables reliable monitoring, analysis, maintenance, and remaining service life prognosis of the product for which the original CAD model no longer applies [1,2], but also provides insight to the designers regarding the product's response to real-world usage [3]. Exponential growth of manufacturing data due to widespread sensor deployment and the emergence of digital twin over the past decade [4] have further highlighted the importance of rapid digitization in fulfilling the promise in creating a *living digital replica* of the physical product, extending its application to the entire product life cycle including process planning and assembly [5].

Within product digitization, reconstructing a 3D model using data obtained from contactless sensors, such as cameras, has attracted broad research interest. In [6], Wang et al. developed a 3D reconstruction method based on the silhouettes of object as projected onto the camera's image plane. The 3D model is represented by a set of "pillars" that underwent a trimming process over images from all view-angles. In [7], photogrammetry was investigated by Galantucci et al. for surface reconstruction. Specifically, RGB images taken from different view-angles are first aligned by matching image features. Point cloud is then obtained using a pair-wise depth map computation algorithm.

Beyond RGB imaging, 3D reconstruction based on laser or infrared pattern has also been reported. In [8], Contri et al. investigated laser-based 3D reconstruction where a planar laser is projected onto the object. The intersecting line between the laser and the surface is then

captured by an RGB camera. The distorted shape of the line provides the basis for reconstruction. In [9], Kinnell et al. developed a stereo camera-based approach for 3D reconstruction. The idea is to project infrared patterns onto the object to help determine its geometry.

Inspired by these prior works and built upon recent development of deep learning, this paper investigates a new 3D modeling method based on neural rendering [10]. The main idea is to use a neural network to model the scene in the form of a *radiance field* where the physical object of interest (e.g., in-service product) and its surroundings reside. In the radiance field, each spatial location has a density

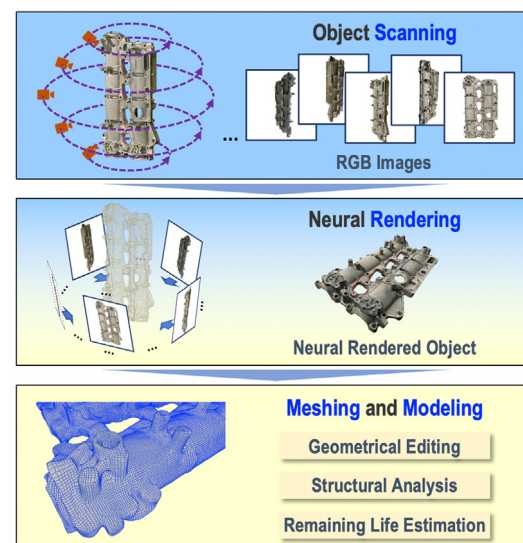


Fig. 1. Workflow of neural rendering-based object 3D modeling.

* Corresponding author.

E-mail address: robert.gao@case.edu (R.X. Gao).

and a view-angle dependent RGB color. Once trained using a set of RGB images, the neural network-modeled radiance field allows not only to distinguish different components in the scene such as air and physical object, but also to reconstruct the scene as viewed from angles not observed in training. Additionally, the encoded color and density information enable generation of textured mesh of the physical object, which can be subsequently imported into a CAD/CAE software to contribute to multiple applications in design and analysis, as shown in Fig. 1. The developed method also addresses some of the limitations in prior works, such as difficulty in handling complex geometry and requiring specialized scanning devices. The effectiveness of the developed method is experimentally evaluated through 3D modeling of a car engine part with complex geometry.

2. Technical background

The method of neural rendering consists of two main parts: 1) radiance field-based scene representation (e.g., physical object and its surroundings) that allows for the generation of view-angle specific RGB images and determine the composition of the scene, and 2) neural network-based modeling of the radiance field.

2.1. Radiance field-based representation

Different from the methods of object representation in prior works, such as intersection of projections and matched image features, radiance field represents a scene as a *function* [11]. The input to the function consists of a spatial location in the scene as represented by Cartesian coordinate $x = (x, y, z)$ and a view-angle r , while its output includes a view-angle dependent RGB color c and a density σ at the location x .

The view-angle r in the function input allows the radiance field to capture color variations of the scene when viewed from different angles (e.g., caused by lighting conditions). The density σ in the output allows the radiance field to distinguish different components in the scene, such as air and physical object, by means of quantifying the transmittance of light when it passes through each spatial location (e.g., percentage of light retained after traveling through each location).

Specifically, for a light that travels from its origin to a location s along its path, the accumulated transmittance along the path follows exponential decay [11]. As a result, the regions of air expect to have low-density values (high transmittance, close to 1), while the regions occupied by the physical object expect to have high-density values (low transmittance, close to 0).

2.2. Neural rendering

To realize radiance field-based scene representation, neural rendering uses a neural network such as multi-layer perceptron (MLP) to iteratively learn the complex radiance field function based on a set of RGB training images, to accurately predict the color and density of the object at each spatial location in the scene [10]. The performance of neural network training is evaluated such that images of the physical scene used for network training can be reconstructed from the

neural rendered scene, which is represented by the network-predicted radiance field.

The procedure is realized that, at each training iteration, radiance field is first predicted by the network. Then, images are taken virtually of the neural rendered scene as represented by the predicted radiance field. To evaluate the quality of the network prediction, these virtual images are taken with the same camera view-angles with which the actual training images were taken and are then compared to the corresponding actual images in a pixel-wise manner. The differences in pixel RGB color resulting from the comparison serve as the basis to update the neural network weights for the subsequent training iteration.

Virtual image-taking of the neural rendered scene follows the imaging physics. In neural rendering, a pinhole camera model is adopted [12]. The idea is to compute the RGB color C of an (virtual) image pixel using the color and density information (in the radiance field) along the ray of light that hits the pixel through the camera's optical center. The computation follows [11]:

$$C = \int_0^D T(s)\sigma(s)c(s)ds, \quad T(s) = \exp\left(-\int_0^s \sigma(s)ds\right) \quad (1)$$

where $c(s)$ is the RGB color at location s on the ray, D is the location where the ray intersects the far-end of the scene, and $T(s)$ is the accumulated transmittance.

Eq. (1) can be viewed as a weighted sum of RGB colors $c(s)$ along the ray. The weight consists of 1) $T(s)$: transmittance over the interval between the location s and the location of the pixel (i.e., $s = 0$), and 2) $\sigma(s)$: density at location s . As a result, the RGB color of the pixel will be dominated by the color of the spatial location on the ray that has both a high transmittance between that location and the pixel (e.g., air) and a high density at that location itself (e.g., solid). In practice, the continuous integral in Eq. (1) is approximated using the sampled locations along the ray.

By alternating 1) prediction of radiance field (by the neural network) and 2) evaluation of the quality of the prediction (using imaging physics) as shown in Fig. 2, the training process iteratively updates the weights of the neural network to minimize the difference between the neural rendered scene and the actual scene, leading to realistic rendering results that serve as the basis for 3D model construction.

2.3. Scene contraction

In many practical settings, the physical object of interest for 3D modeling is located in a large or even unbounded background environment (e.g., on the shop floor or in the field). In these cases, a ray of light travel a large or even infinite distance to intersect with the scene boundary at the far-end, making spatial location sampling along the ray and computation of Eq. (1) challenging. Considering that the background in the scene is usually not of interest for 3D modeling, the method of contraction is investigated to “warp” the background environment into a small, finite space using the contraction equation [13]:

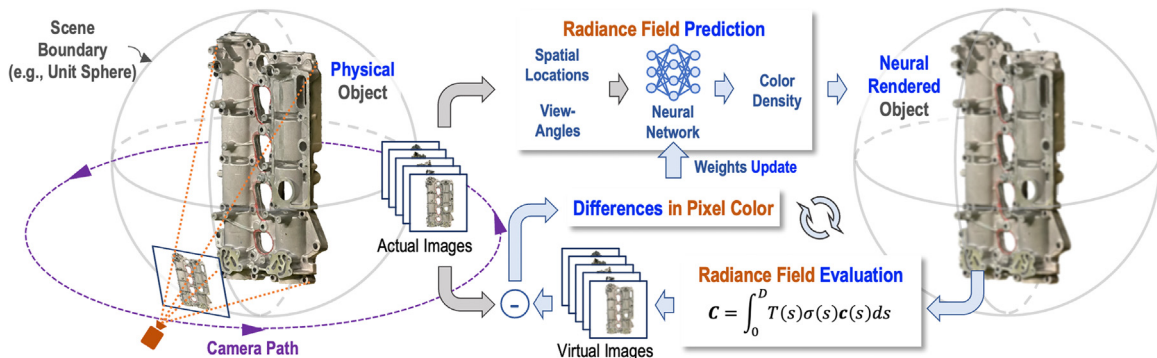


Fig. 2. Network training process for neural rendering of a physical object.

$$f(x) = \begin{cases} x & \|x\| \leq 1 \\ \left(2 - \frac{1}{\|x\|}\right) \left(\frac{x}{\|x\|}\right) & \|x\| > 1 \end{cases} \quad (2)$$

Eq. (2) separates the scene into two regions: 1) the spatial locations inside a sphere of radius 1 that encompasses the physical object, and 2) the locations outside of this unit sphere (e.g., background). When sampling along the rays for neural rendering, Eq. (2) does not modify the locations inside the unit sphere while contracting the entire background environment into a sphere of radius 2. As a result, the rays of light and the far-end of the scene always intersect at the surface of the sphere of radius 2.

3. Experimental evaluation

The performance of the developed neural rendering technique is evaluated using a cylinder head with complex geometry from a car engine. The cylinder head is mounted on a steel stand located at the center of the workspace as shown in Fig. 3. Images used to train the neural network are taken by the camera of an iPhone 12 attached to the end-effector of a Kinova Gen2 robot, which is mounted on a Robotnik Summit-XL (mobile) robot.

During the experiment, the movement of the Summit-XL around the stand, as denoted by the orange-colored trajectory, allows the part to be scanned at different longitudes. Also, change of the end-effector's pose of the Kinova robot allows the part to be scanned at varying latitudes. Specifically, a total of 12 different longitudes with an equal increment of 30° around the stand and 9 different latitudes with an equal increment of 20° from −80 to 80° are used. Both Robotnik and Kinova robots are controlled by an open architecture robot operating system (ROS)-integrated computer via Wi-Fi based command transmission.

For purpose of illustration, three camera trajectories are shown in Fig. 3 as yellow dashed lines. In total, 108 images are taken at a resolution of 3024 × 3024 to serve as training data. In Fig. 3, three training images taken from the camera locations ①, ②, and ③ are illustrated at the bottom of the figure.

Neural rendering and mesh generation of the workspace are carried out using the cloud-based software by LumaLabs [14]. Using full-resolution images for training, the total time for neural rendering and mesh generation is around 1.5 h. This represents potentially significant time reduction as compared to manual 3D model creation. The mesh of the cylinder head is first extracted from the scene using a bounding box that

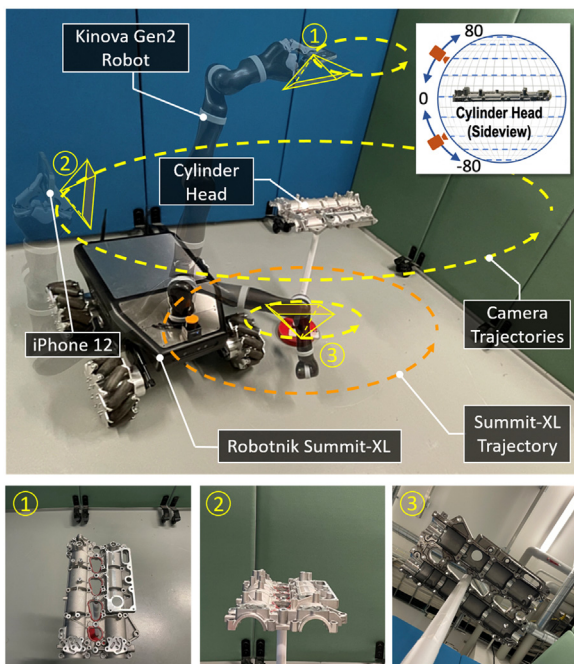


Fig. 3. Experiment workspace and sample training images.

encapsulates the cylinder head. The extracted mesh is pre-processed by Recap Photo for artifacts removal and mesh conversion before being imported into Fusion 360 to generate surface and solid models for applications such as geometrical editing and structural analysis.

4. Results discussion

4.1. Neural rendered scene

The result of neural rendering of the cylinder head and its surroundings are evaluated using the images reconstructed from the radiance field as viewed from angles that are not used during the training process. An example is shown in Fig. 4.

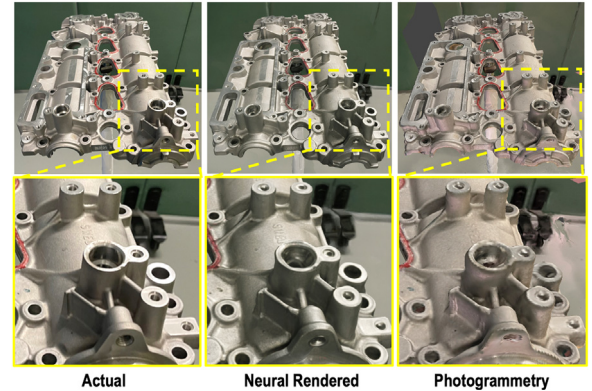


Fig. 4. Comparison between actual and reconstructed images.

It is noted that neural rendering has achieved a good level of photorealism. The colors associated with different surfaces of the cylinder head, such as the red edge of the openings for plug installation, have been properly shown. Furthermore, the learned radiance field has correctly captured the geometrical relationship among different bodies on the cylinder head, such as the set of extruded cylinders with varying heights and sizes. This leads to rendering results that are geometrically consistent with the actual image. Additionally, geometrical consistency is maintained with respect to the surroundings (e.g., proper occlusion of dividers).

To quantify the faithfulness of the neural rendered image compared to the actual image, the structural similarity index measure (SSIM) is evaluated [15]. SSIM compares two images by first segmenting each into a grid of smaller windows. For each pair of corresponding windows from the two images, a composite index is computed based on the statistics of pixels within the window. Then, composite index is averaged over all window pairs. An SSIM of 0.694 is obtained for the neural rendered image. In comparison, image reconstructed using photogrammetry, which is a widely used 3D reconstruction method that is also based on RGB images [16], has achieved an SSIM of 0.559. This demonstrates the improvement of neural rendering in realizing photorealistic image reconstruction.

4.2. Surface meshing and CAD modeling

The geometry of the cylinder head generated by neural rendering is represented using triangular mesh (tri-mesh), which has also been compared to the outcome from photogrammetry, as shown in Fig. 5. It is noted that neural rendering performed better than photogrammetry in capturing details in the concave geometrical regions, such as the mounting platform in region A, opening in region B, and transition area in region C of the part. To enable editing in CAD/CAE software, the tri-mesh is converted to quad-mesh to establish a direct correlation between the mesh and freeform surface models, such as T-spline [17], as shown in Fig. 6.

The geometrical editing functionality is demonstrated in Fig. 7. For illustration purpose, the tri-mesh from neural rendering is first simplified by adaptively reducing the number of faces, before converting

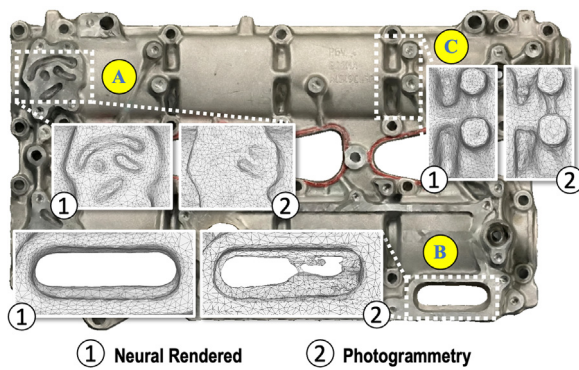


Fig. 5. Mesh comparison between neural rendering and photogrammetry.

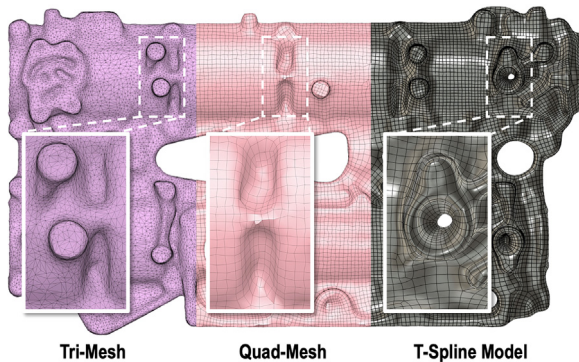


Fig. 6. Conversion from tri-mesh to T-spline surface model.

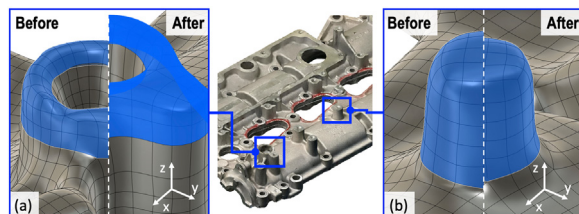


Fig. 7. Examples of geometrical editing of neural-rendered cylinder head.

into T-spline model. The selected surface regions for modification are highlighted in blue. The surface is scaled up in the z direction by a factor 1.2, as shown in Fig. 7(a), while scaled down in the (x; y; z) directions by a factor of 0.95, as shown in Fig. 7(b). With the editing functionality, designers will be able to directly work on top of the neural rendered product for design improvement, without having to build the 3D model from scratch.

To enable further functionalities such as structural analysis or design optimization, the T-spline surface model is converted to a B-rep solid model. The solid model can be acted upon for in-service product performance analysis, remaining service-life estimation, feedback to design optimization, etc.

It should be noted that neural rendering has several limitations. As with other RGB image-based methods, it is limited in reconstructing geometry and features that are under-illuminated, such as deep pockets and small holes. Additionally, it cannot capture the internal part structure as is the case with all non-penetrating scanning methods. Therefore, it is suited for surface reconstruction only. Furthermore, realizing its full potential for product digitization also requires the development of new algorithms for efficient parametric surface generation from large-scale tri-mesh to facilitate surface geometric editing.

5. Conclusions

A novel method for rapid digitization of physical objects is presented to support the analysis of in-service products where an up-to-

date CAD model is not available. The method represents the physical object and the surroundings as a scene by means of radiance field, which encodes the color and density information that describe its composition. Guided by imaging physics, a neural network is trained using RGB images to model the radiance field for object modeling and mesh generation. Experimental evaluation using a cylinder head from a car engine confirms the following contributions from the neural rendering-based method:

- Photorealistic and geometrically consistent 3D modeling of in-service products even from view-angles not seen in training, without requiring specialized scanning devices.
- Mesh/model generation suited for design applications such as geometrical editing and structural analysis.
- Potentially significant time reduction as compared to manual 3D model creation of in-service products.

Future effort will be directed to improving the performance of neural rendering in capturing object details and conversion to parametric models. Its utility in developing digital twins for human-robot collaborative assembly and hybrid autonomous manufacturing will be investigated.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Gao and Zhang acknowledge support from the National Science Foundation under grant CMMI-1830295 and the Engineering Research Center (ERC-HAMMER) grant EEC-2133630.

References

- [1] Erdős G, Nakano T, Váncza J (2014) Adapting CAD Models of Complex Engineering Objects to Measured Point Cloud Data. *CIRP Annals* 63(1):157–160.
- [2] Qie Y, Bickel S, Wartzack S, Schleich B, Anwer N (2021) A Function-Oriented Surface Reconstruction Framework for Reverse Engineering. *CIRP Annals* 70(1):135–138.
- [3] Stark R, Grosser H, Müller P (2013) Product Analysis Automation for Digital MRO Based on Intelligent 3Ddata Acquisition. *CIRP Annals* 62(1):123–126.
- [4] Gao R, Wang L, Helu M, Teti R (2020) Big Data Analytics for Smart Factories of the Future. *CIRP Annals* 69(2):668–692.
- [5] Liu S, Wang XV, Wang L (2022) Digital Twin-Enabled Advance Execution for Human-Robot Collaborative Assembly. *CIRP Annals* 71(1):25–28.
- [6] Wang L, Mohammed A, Onori M (2014) Remote Robotic Assembly Guided by 3D Models Linking to a Real Robot. *CIRP Annals* 63(1):1–4.
- [7] Galantucci LM, Pesce M, Lavecchia F (2015) A Stereo Photogrammetry Scanning Methodology, for Precise and Accurate 3D Digitization of Small Parts with Sub-millimeter Sized Features. *CIRP Annals* 64(1):507–510.
- [8] Contri A, Bourdet P, Lartigue C (2002) Quality of 3D Digitised Points Obtained with Non-contact Optical Sensors. *CIRP Annals* 51(1):443–446.
- [9] Kinnell P, Rymer T, Hodgson J, Justham L, Jackson M (2017) Autonomous Metrology for robot Mounted 3D Vision Systems. *CIRP Annals* 66(1):483–486.
- [10] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2020) NeRF: representing Scenes as Neural Radiance Fields for View Synthesis. *Computer Vision – ECCV 2020*, 405–421.
- [11] Max N, Chen M (2005) *Local and Global Illumination in the Volume Rendering Integral* (No. UCRL-PROC-216495), Lawrence Livermore National Laboratory (LLNL) Livermore, CA, (United States).
- [12] Tsai R (1987) A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology using off-the-Shelf TV Cameras and Lenses. *IEEE Journal on Robotics and Automation* 3(4):323–344.
- [13] Barron JT, Mildenhall B, Verbin D, Srinivasan PP, Hedman P (2022) Mip-nerf 360: unbounded Anti-aliased Neural Radiance Fields. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.
- [14] LumaLabs, <https://lumalabs.ai>
- [15] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13(4):600–612.
- [16] Schonberger JL, Frahm JM (2016) Structure-from-Motion Revisited. *IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113.
- [17] Sederberg TW, Zheng J, Bakenov A, Nasri A (2003) T-splines and T-NURCCs. *ACM Transactions on Graphics (TOG)* 22(3):477–484.