

Viewpoint Selection for Texture Reconstruction with Inverse Rendering

Vadim Sanzharov¹, Vladimir Frolov^{1,2}

¹Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia

²Keldysh Institute of Applied Mathematics RAS, Miusskaya pl., 4, Moscow, 125047, Russia

Abstract

Viewpoint selection methods have a variety of applications in different fields of computer graphics and computer vision, including shape retrieval, scientific visualization, image-based modeling and others. In this paper we investigate the applicability of existing viewpoint selection methods to the problem of textures reconstruction using inverse rendering. First, we use forward rendering to produce path-traced images of a textured object. Then we apply different view quality metrics to select a set of images for texture reconstruction. Finally, we perform material and texture reconstruction using these image sets and evaluate the quality of the results. We show that using viewpoint selection methods allows to achieve faster inverse rendering times while maintaining quality of the results.

Keywords

inverse rendering, viewpoint selection, 3d reconstruction, texture reconstruction.

1. Introduction

Viewpoint selection has many different applications in computer graphics and computer vision. For example, in scientific visualization it is important to produce images conveying the most information about the subject. Simultaneous localization and mapping (SLAM) problem requires planning motion of a robotic system which can provide the best views for construction of an accurate map.

Intuitively viewpoint selection can be formulated as a simple question - "which views of a 3D scene/model are the best?" The measure, which determines how good is a specific view, depends on the problem.

Inverse rendering aims to recover a 3D scene, including geometry, lighting and materials, given a set of its images. Most approaches to inverse rendering are based on differential rendering [1], which involves using gradient-based optimization to iteratively improve input parameters (such as vertex positions in 3D models, intensity of light sources, material properties, etc.) so rendered images match input images (for example, photographs). Inverse rendering problem can also be solved without gradient-based methods for specific cases [2] or with some simplifications and a

GraphiCon 2023: 33rd International Conference on Computer Graphics and Vision, September 19–21, 2023,

V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

✉ vadim.sanzharov@graphics.cs.msu.ru (V. Sanzharov); vfrolov@graphics.cs.msu.ru (V. Frolov)

ORCID [0000-0001-6455-6444](https://orcid.org/0000-0001-6455-6444) (V. Sanzharov); [0000-0001-8829-9884](https://orcid.org/0000-0001-8829-9884) (V. Frolov)

© 2023 Copyright for this paper by its authors.

 Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

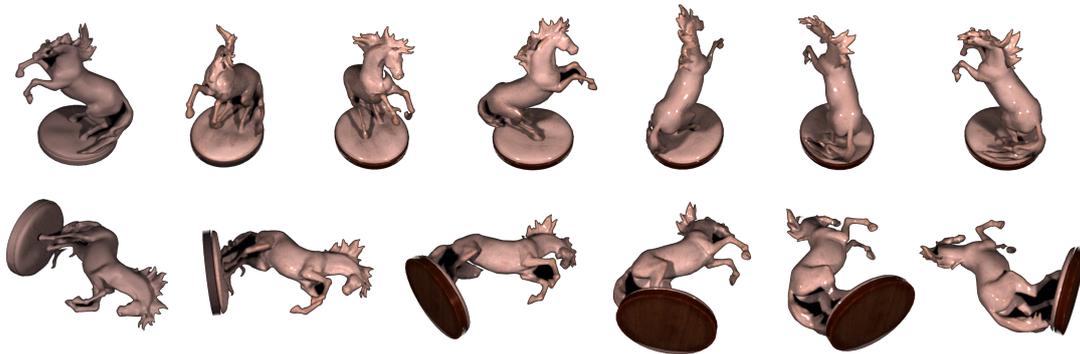


Figure 1: Several example views of a 3D model in two positions on the virtual turntable. Texture optimization iterations increase left to right.

priori knowledge [3]. As differential rendering is more widely applicable, further on in this paper we will assume it as an approach of choice for texture reconstruction.

The surface of real-world objects is generally non-uniform and has various peculiarities. Therefore, material and its surface reflectance properties often need to be modeled as spatially-varying. In practice, this means that material reconstruction process uses textures as parameters of a material model. To accurately and fully recover a texture of a 3D object, inverse rendering system should be given a set of views of that object covering the whole surface. As a simplest example consider a 3D model of a cube. To reconstruct a texture of a cube we will need to have views of all of the sides of this cube. And if one of the sides is never visible in any of the views, there is no information to reconstruct the corresponding part of a texture. If the said cube has a material with non-negligible specular reflective properties, several images of the same side could be required. The reason is that visual manifestation of specular features of the surface depends on the viewing angle.

It is also should be noted that inverse rendering is a computationally costly task. And one of the factors making it such is the need to reconstruct textures, which means having high-dimensional output as well as input. If we don't want to sacrifice visual quality of the result by reducing reconstructed textures resolution, another possibility to speed up computations (as well as capturing real-world data) would be to reduce the number of input images.

Thus, for material reconstruction using differential rendering we need to have a set of views of a target 3D object such that every polygon would be visible at least in a single view. At the same time we would like to keep the number of views to a minimum to reduce computational and image capturing workloads.

In this paper we propose to apply viewpoint selection methods to produce an input image set for material (texture) reconstruction with inverse rendering. We verify our proposition on synthetic data using Mitsuba3 [4] as a basis for the inverse rendering process.

2. Related work

Research on viewpoint selection yielded a number of different measures, which were surveyed in [5, 6]. In [5] measures were classified according to data used to compute them - visible polygons projected area, silhouette, depth, stability and curvature. Authors also evaluate performance of all surveyed measures (22 in total) against benchmark proposed in [7]. In [6] survey viewpoint selection methods are classified into three groups - based on geometric information, based on visual features (predominantly using mesh saliency) and semantic-based (using semantic segmentation of 3D models).

Apart from measures covered in [5] there are deep learning approaches [8, 9, 10], which usually aim to select views based on saliency and human perception.

Among the variety of existing approaches we will consider those that are based on geometric information. In [5] authors review applications of different measures and geometry-based ones found use in a variety of different problems. Specifically in image-based modeling and rendering which can be considered as the most similar to the stated problem of inverse rendering. For example, in [11] authors minimize the number of views for 3D reconstruction with laser scanning and in [12] - the number of images needed to represent a scene for image-based rendering. Additionally, methods based on saliency and human perception seem to be conceptually not well-suited for inverse rendering problem, since they tend to select views containing so-called "points of interest" and highly detailed areas. However, in the stated problem of texture reconstruction it is necessary to explore the entire mesh surface (or at least visible in the available set of views) to produce the most complete and accurate texture.

One of the earliest works on viewpoint selection proposes using angle between viewing direction and surface normals of a 3D model [13] as a measure. In [14] authors additionally incorporate total number of visible polygons and their projected area as a viewpoint quality measure, weighing contributions from different factors. However, it's unclear how to determine good weights.

Several measures use projected area of polygons to compute information theoretic quantities including entropy [15], relative entropy (Kullback–Leibler divergence) [16] and mutual information [17, 18]. Among these measures projected area entropy [15] and mutual information [17] found the most applications in different fields as reported in [5]. It is also worth noting that in [12] authors propose an algorithm for selecting a set of views covering all polygons using projected area entropy [15].

In [19] authors propose several new entropy-based measures using quantities other than projected area for its computation, including depth, field data and shading coefficients. This work deals specifically with scientific visualization applications.

In [20] depth variation and depth distribution entropy are used for selecting best views to demonstrate 3D models of museum pieces. Authors also propose an algorithm for selecting a set of views. However, their goal is to select a small representative set of views (in particular, four views) best suited for human perception. Which does not necessarily coincide with the stated goal of view selection in inverse rendering where every polygon should be visible at least in a single view.

In our work we implement viewpoint entropy [15] and viewpoint mutual information [17]. We also propose a new measure which uses texture area covered by polygons.

3. Proposed method

3.1. Viewpoint quality measures

First base measure we implement is viewpoint entropy [15, 12]. It is based on the projected area of a 3D scene given a particular viewpoint and is defined as follows:

$$VE(v) = - \sum_{i=0}^N \frac{a_z(v)}{a_t(v)} \log \frac{a_z(v)}{a_t(v)} \quad (1)$$

where N is the number of polygons in the 3D model, $a_z(v)$ is projected area of polygon z from viewpoint v and $a_t(v)$ is projected area of the whole 3D model from viewpoint v .

Ratio $\frac{a_z(v)}{a_t(v)}$ is proportional to the cosine of the angle between normal to polygon z and the inverse viewing direction (thus, it incorporates measure proposed in [13]). At the same time, this ratio is inversely proportional to the squared distance between the camera and polygon z . So, $\frac{a_z(v)}{a_t(v)}$ will increase as polygon z gets closer to the camera and as polygon's normal aligns with the inverse viewing direction.

Another base measure we use in our experiments is viewpoint mutual information (VMI) [17]:

$$p(z) = \sum_{v \in V} \frac{a_t(v)}{\sum_{v \in V} a_t(v)} * \frac{a_z(v)}{a_t(v)} \quad (2)$$

$$VMI(v) = \sum_{i=0}^N \frac{a_z(v)}{a_t(v)} \log \left[\frac{a_z(v)}{a_t(v)} / p(z) \right] \quad (3)$$

where V is the set of all views.

Authors of this viewpoint quality measure [17] state that VMI will have higher values for views which are highly coupled with, for example, a small number of polygons with low average visibility. And lowest values correspond to representative views. But in our case we are actually interested in finding views which would cover all polygons in the model and selecting views which contain polygons not visible from other viewpoints can be beneficial. Thus, we assume higher VMI values as better for our purpose.

As a modification of (1) we propose "texture area entropy" viewpoint quality measure:

$$VE_{texture}(v) = - \sum_{i=0}^{N_v} \frac{b_z}{b_t} \log \frac{b_z}{b_t} \quad (4)$$

where b_z is the area of the texture (which is being reconstructed) that is used by polygon z , b_t is the total texture area used by polygons of the model and N_v is the number of polygons visible from viewpoint v .

This measure will have a higher value for views with larger visible texture area. Thus with texture area entropy we aim to select views that contribute the most to the texture we want to reconstruct with inverse rendering. Note that texture area may not correlate with polygon area as it depends entirely on texture coordinate parametrization of the 3D model. Since b_z and b_t do not depend on a view, their values can be computed only once.

3.2. Viewpoint selection algorithm

For accurate texture reconstruction we need a set of views of a target 3D object such that every polygon would be visible at least in a single view. However, we must take into account that a set of possible views is defined by image acquisition setup. Real camera has its position and orientation (relative to the 3D object being captured) restricted, unlike virtual camera which has no physical size. Thus, we formulate the problem as follows: select a subset of N views from the set of all available M views such that each polygon z that is visible in at least one view of M , is also visible in at least one view of N .

To solve it we use an algorithm based on [12]. We select the best view depending on the quality measure being used and mark all polygons visible in that view as visited. Then we compute measures again excluding all visited polygons and continue the process until all polygons are visited.

Algorithm 1: Selection of N views

Data: V - set of all views, Z - set of all polygons visible in V , VQ - quality measure

Result: M - subset of views

```

1  $Q \leftarrow \emptyset$ ;
2  $M \leftarrow \emptyset$ ;
3 while  $Q \neq Z$  do
4    $measures \leftarrow \emptyset$ ;
5   forall  $v \in V$  do
6     forall  $z \in Z$  do
7       if  $z \notin Q$  then
8          $m \leftarrow \text{ComputeQualityMeasure}(VQ, v, z)$ ;
9          $measures[v] \leftarrow m$ ;
10      end
11    end
12  end
13   $view \leftarrow \text{GetBestView}(VQ, measures)$ ;
14  forall  $z \in Z$  do
15    if  $z$  is visible in view then
16       $Q \leftarrow z \cup Q$ ;
17    end
18  end
19   $M \leftarrow view \cup M$ 
20 end

```

For viewpoint entropy measure the condition in line 7 for the above algorithm is changed to include only polygons with big enough relative projected area (as proposed in [12]):

$$z \notin Q \wedge 100 * \frac{a_z}{max_area_z} > 90 \quad (5)$$

where a_z is projected area of polygon z and max_area_z is maximum projected area of polygon z

among the set of all views V .

3.3. Rendering pipeline

To evaluate the proposed approach we generate synthetic data with forward rendering and then use it for inverse rendering. Our rendering pipeline is built around Mitsuba3 rendering system[4]. The scheme of the pipeline is presented in fig. 2

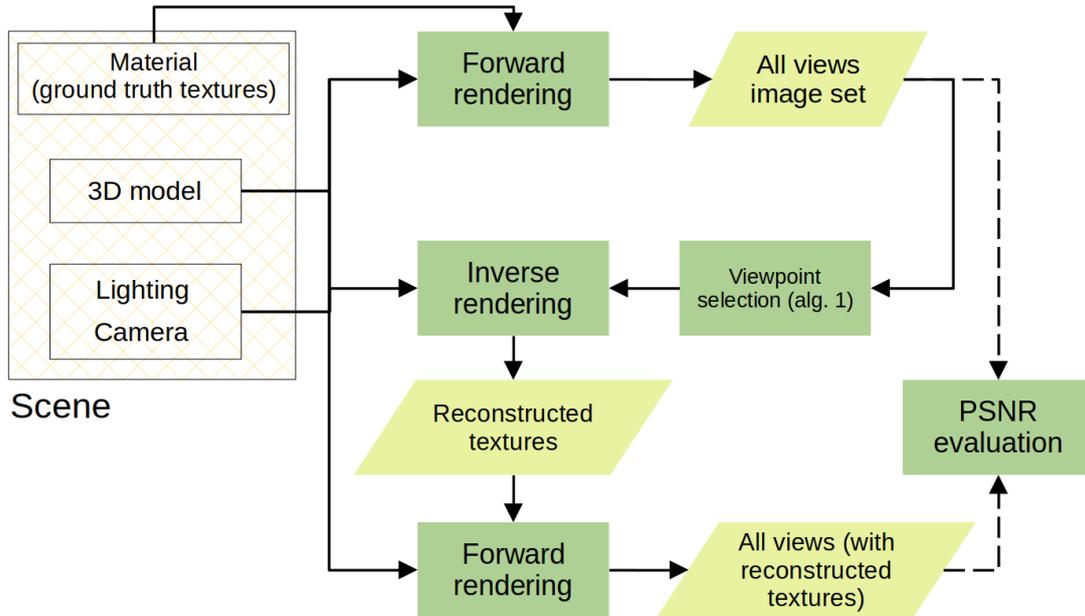


Figure 2: Rendering pipeline scheme used to perform experimental evaluation. 3D model is rotated 360 degrees with 10 degree step in two positions yielding 72 views in total.

Our 3D scene recreates a simple real-life image acquisition setup with a 3D object placed in the center of a turntable (fig. 1) and camera looking at it from above at around 45 degrees angle. We first render a set of images representing all possible views of a 3D model which consists of 72 images - 360 degrees rotation with 10 degree step for 2 positions of a model on the turntable (standing and lying on one of the sides). Forward rendering is conducted with known material model (including textures) and lighting conditions.

For the experimental evaluation we chose three 3D models of increasing complexity (fig. 3). Material models are lambertian diffuse with single "reflectivity" texture parameter and principled model with "base_color" and "roughness" texture parameters.

Next, a subset of all possible views is chosen using algorithm 1 and passed on to the inverse rendering stage. To isolate the problem of texture reconstruction we set position and orientation of the camera and the 3D model as they were during the forward rendering stage. Lighting conditions also stay the same. Textures are initialized as uniform gray and act as optimization target for the inverse rendering stage.

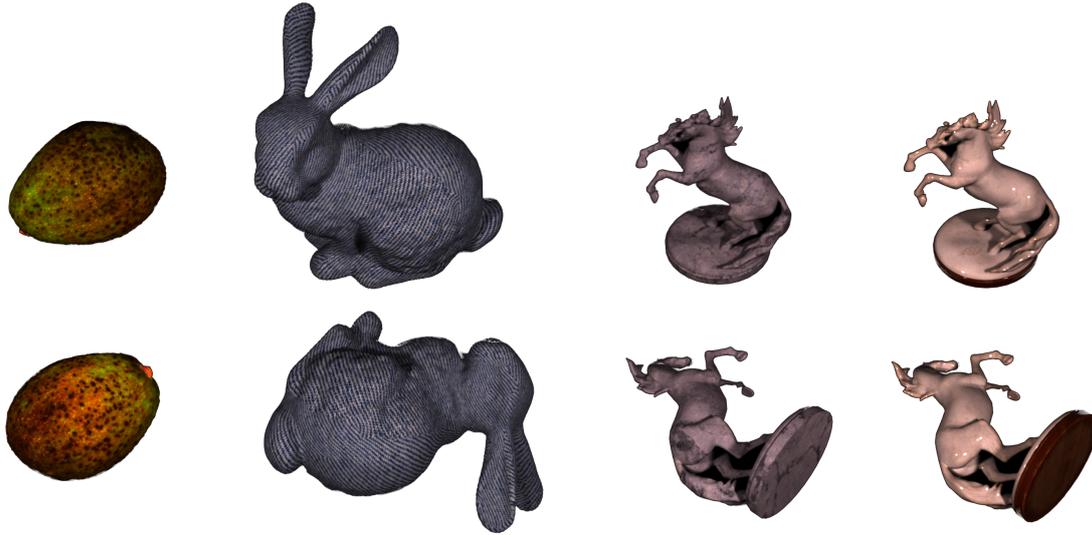


Figure 3: 3D models used for testing in two positions on the virtual turntable. Horse model was rendered with diffuse material and principled material with glossy specular component.

Inverse rendering is performed using Mitsuba3 "reparameterized direct integrator". As a loss function for diffuse material we use L_1 distance between target and rendered images clamped to $(0, 1)$. And for principled material with glossy specular reflections we use L_1 distance with log-encoding pixels as in [21] to reduce the influence of highlights. We also add smoothing regularization term for textures which are being optimized:

$$\Lambda = \lambda * \sum_{i,j} (|\alpha[i + 1, j] - \alpha[i, j]| + |\alpha[i, j + 1] - \alpha[i, j]|) \quad (6)$$

where $\alpha[i, j]$ is the value of the (i, j) pixel in the texture and λ is some small constant value (around 0.01 in our case with texture resolution equal to 1024×1024).

Inverse rendering stage outputs optimized textures which are then used to perform forward rendering of all 72 views of a 3D model once again. Finally, these rendered images with reconstructed textures are compared to images obtained during the first forward rendering stage with ground truth textures.

3.4. Results and discussion

Described pipeline was executed 12 times - for all 4 scenes (fig. 3) and each quality measure (VE , $VE_{texarea}$, VMI). In each scenario we measured PSNR for all views (table 1), inverse rendering time (fig. 7) and size of the subset of images produced by viewpoint selection algorithm (table 2).

Results show that viewpoint measures demonstrate PSNR values comparable to each other in most cases. Only notable differences are lower values for $VE_{texarea}$ on *avocado* model and for VMI on *bunny* model. On *avocado* model $VE_{texarea}$ shows PSNR values smaller than other measures, but still around 50 in the worst case. Resulting textures and rendered images are almost

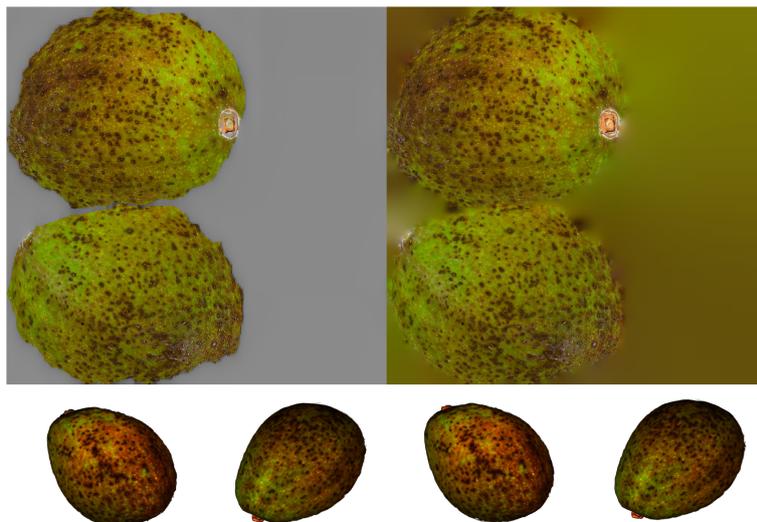


Figure 4: Left - texture reconstructed from views selected by $VE_{texarea}$ measures and algorithm 1, right - ground truth texture.

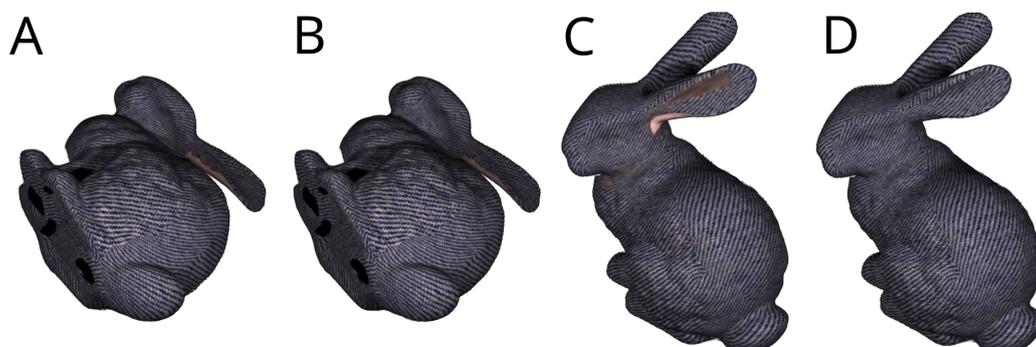


Figure 5: Bunny model rendered with A) texture reconstructed on views selected with $VE_{texarea}$ measure and algorithm 1; B) texture reconstructed on all views; C) texture reconstructed on views selected with VMI measure and algorithm 1; D) texture reconstructed from all views

identical (fig. 4). On *bunny* model low PSNR value for VMI measure is likely caused by the fact that some parts of the model (in particular, ears) are not well lit in all views. This leads to appearance of non-reconstructed areas of the target texture even when all 72 views were used (fig. 5, A and B). Because views with well lit ear parts of the model were not chosen with VMI measure the target texture had significant missing areas in that part (fig. 5, C and D).

For the horse model all viewpoint selection measures produce large subsets of images (75% of all views) but at the same time achieve high PSNR values which are also slightly better than using all views for reconstruction (table 1, fig. 6) while still saving time. Visualized differences between images rendered with reconstructed texture and ground-truth texture (fig. 6) show that largest differences appear mostly in the shadowed areas of the model and at the bottom part of the base even when using all views. This is consistent with the results for the bunny model - not

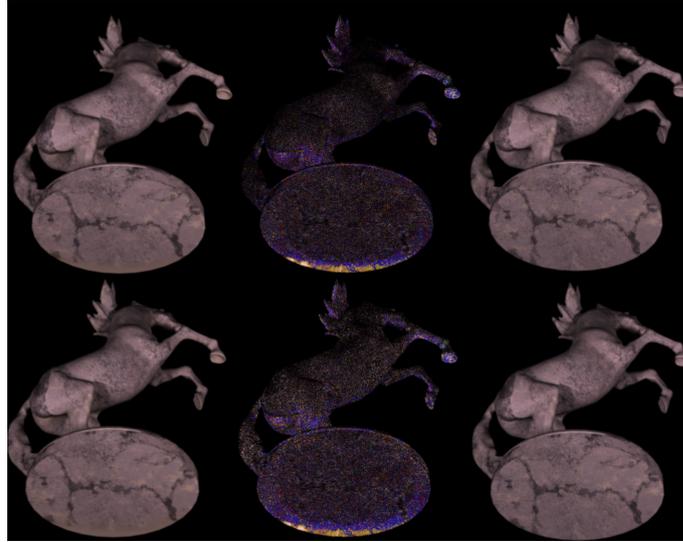


Figure 6: View with the lowest PSNR value for the horse model with diffuse material. Top row: horse model rendered with texture reconstructed on views selected with proposed $VE_{texarea}$ measure and algorithm 1 (left), ground-truth texture (right), image difference $\times 10$ (center). Bottom row: horse model rendered with texture reconstructed on all 72 views (left), ground-truth texture (right), image difference $\times 10$ (center).

well lit parts of the model are badly reconstructed. Also note that in the example shown on fig. 6 difference for all views is larger than for smaller number of views selected with the proposed measure.

In case of glossy specular material time saves are more significant as inverse rendering is more computationally expensive in this case (we optimize two textures). Thus, for more complex material models and more optimized parameters using viewpoint selection methods may be especially beneficial. Rendered images for different iterations of reconstruction process with the proposed measure for glossy material are shown on fig. 1.

Overall, using viewpoint selection measures allows to reduce number of images for texture reconstruction by 25-89% (table 2) depending on the complexity of the 3D model. All three measures we implemented show very similar reconstruction quality (table 1) with VE being the most consistent. We also compared measures using quantity incorporating both quality and time (table 3). It clearly shows that in every case using VE or $VE_{texarea}$ is much better than just using all available views with VE being the best measure overall. VMI is worse than other tested measures, losing to all views in the case which was described earlier.

4. Conclusion

Our goal was to demonstrate feasibility of using viewpoint selection methods in the field of inverse rendering. To do this we implemented two widely used viewpoint quality measures and also proposed a new one. We carried out experimental evaluation on several 3d models with

Table 1

Average PSNR and standard deviation for different viewpoint quality measures. PSNR is computed between view rendered with reconstructed textures and with ground truth textures (fig. 2)

Scene	VE	$VE_{texarea}(ours)$	VMI	All views
avocado	56.2 ± 3.1	54.8 ± 4.9	56.5 ± 4.0	59.2 ± 2.7
bunny	49.4 ± 3.4	49.4 ± 3.4	44.5 ± 7.2	51.9 ± 2.6
horse-diffuse	56.6 ± 2.4	56.6 ± 2.4	56.6 ± 2.4	56.5 ± 2.8
horse-specular	45.0 ± 3.4	45.0 ± 3.3	45.0 ± 3.3	45.2 ± 3.2

Table 2

Number of views selected for different viewpoint quality measures

Scene	VE	$VE_{texarea}(ours)$	VMI	All views
avocado	8	8	10	72
bunny	31	31	31	72
horse-diffuse	54	55	54	72
horse-specular	54	55	54	72

Table 3

Comparison of viewpoint quality measures by quality and performance, $\frac{1}{MSE*time}$, where MSE is mean squared error between image rendered with reconstructed texture and image rendered with ground truth texture.

Scene	VE	$VE_{texarea}(ours)$	VMI	All views
avocado	390.05	283.91	363.88	91.87
bunny	21.95	20.24	6.93	16.59
horse-diffuse	60.68	62.48	59.03	50.05
horse-specular	3.63	3.58	3.54	2.84

different materials. From the experiments we found that viewpoint selection methods can be used to reduce inverse rendering execution time by 20-89% depending on complexity of a 3D model (fig. 7) in turntable-based image acquisition setup. At the same time, reconstruction quality is maintained in most cases, special care needed for models with parts which are not well lit in most views. All measures we tested show similar performance with viewpoint entropy (VE) achieving consistently good results in all cases (table 3). Our measure $VE_{texarea}$ is a close second and has a small advantage in computational costs of the measure itself - data used by it needs to be computed only once for the whole process of selecting N views, while for VE everything needs to be recalculated N times. We assume that proposed measure $VE_{texarea}$ achieves better results for cases with complex geometry and prominent texture as $VE_{texarea}$ is directly dependent on texture mapping of a 3D model.

As future work, experiments with real data are needed to further prove viability of viewpoint selection methods in inverse rendering. It would also be interesting to explore other viewpoint quality measures. Especially those that incorporate information on how well lit is the particular

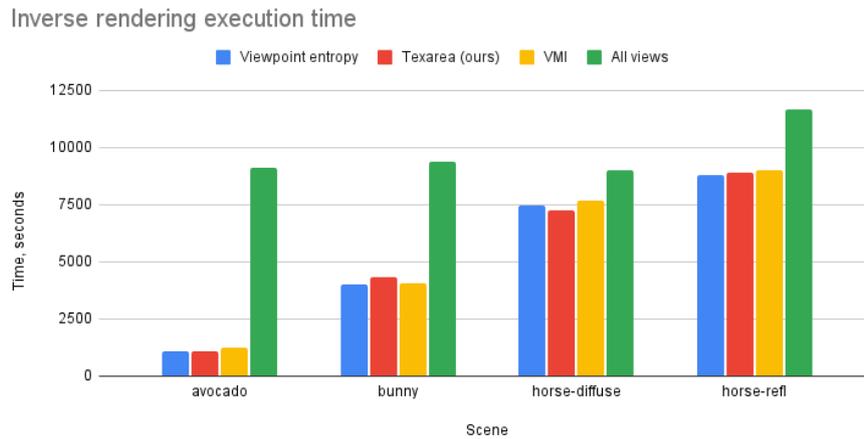


Figure 7: Inverse rendering execution time for different viewpoint quality measures. RTX 2070 Super, 150 iterations, rendering resolution 1024×1024 , target texture resolution 1024×1024 .

polygon. Another possible research direction is associated with exploring how several different views of the same polygon influence reconstruction of reflective materials.

Acknowledgments

The work was sponsored by the non-profit Foundation for the Development of Science and Education "Intellect".

References

- [1] S. Zhao, W. Jakob, T.-M. Li, Physics-based differentiable rendering: from theory to implementation, in: ACM siggraph 2020 courses, 2020, pp. 1–30. doi:10.1145/3388769.3407454.
- [2] S. Zhao, F. Luan, K. Bala, Fitting procedural yarn models for realistic cloth rendering, ACM Transactions on Graphics (TOG) 35 (2016) 1–11. doi:10.1145/2897824.2925932.
- [3] Z. Zhou, G. Chen, Y. Dong, D. Wipf, Y. Yu, J. Snyder, X. Tong, Sparse-as-possible svbrdf acquisition, ACM Transactions on Graphics (TOG) 35 (2016) 1–12. doi:10.1145/2980179.2980247.
- [4] W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, Z. Zhang, Mitsuba 3 renderer, 2023. URL: <https://mitsuba-renderer.org>.
- [5] X. Bonaventura, M. Feixas, M. Sbert, L. Chuang, C. Wallraven, A survey of viewpoint selection methods for polygonal models, Entropy 20 (2018) 370. doi:10.3390/e20050370.
- [6] Y. Zhang, G. Fei, Overview of 3d scene viewpoints evaluation method, Virtual Reality & Intelligent Hardware 1 (2019) 341–385. doi:10.1016/j.vrih.2019.01.001.
- [7] H. Dutagaci, C. P. Cheung, A. Godil, A benchmark for best view selection of 3d objects, in:

- Proceedings of the ACM workshop on 3D object retrieval, 2010, pp. 45–50. doi:10.1145/1877808.1877819.
- [8] R. Song, W. Zhang, Y. Zhao, Y. Liu, Unsupervised multi-view cnn for salient view selection and 3d interest point detection, *International Journal of Computer Vision* 130 (2022) 1210–1227. doi:10.1007/s11263-022-01592-x.
- [9] S.-h. Kim, Y.-W. Tai, J.-Y. Lee, J. Park, I. S. Kweon, Category-specific salient view selection via deep convolutional neural networks, in: *Computer Graphics Forum*, volume 36, Wiley Online Library, 2017, pp. 313–328. doi:10.1111/cgf.13082.
- [10] R. Song, Y. Liu, P. L. Rosin, Distinction of 3d objects and scenes via classification network and markov random field, *IEEE Transactions on Visualization and Computer Graphics* 26 (2018) 2204–2218. doi:10.1109/TVCG.2018.2885750.
- [11] N. A. Massios, R. B. Fisher, A best next view selection algorithm incorporating a quality criterion, in: *British Machine Vision Conference*, 1998, pp. 780–789.
- [12] P.-P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, Automatic view selection using viewpoint entropy and its application to image-based modelling, in: *Computer Graphics Forum*, volume 22, Wiley Online Library, 2003, pp. 689–700. doi:10.1111/j.1467-8659.2003.00717.x.
- [13] T. Kamada, S. Kawai, A simple method for computing general position in displaying three-dimensional objects, *Computer Vision, Graphics, and Image Processing* 41 (1988) 43–56. doi:10.1016/0734-189X(88)90116-8.
- [14] D. Plemenos, M. Benayada, Intelligent display techniques in scene modelling. new techniques to automatically compute good views, in: *Proceedings of the International Conference GraphiCon'96*, Saint Petersburg (Russia), 1996.
- [15] P.-P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, Viewpoint selection using viewpoint entropy, in: *Proceedings of the Vision Modeling and Visualization Conference 2001*, 2001, pp. 273–280.
- [16] L. Neumann, M. Sbert, B. Gooch, W. Purgathofer, et al., Viewpoint quality: Measures and applications, in: *Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, 2005, pp. 185–192.
- [17] M. Feixas, M. Sbert, F. González, A unified information-theoretic framework for viewpoint selection and mesh saliency, *ACM Transactions on Applied Perception (TAP)* 6 (2009) 1–23. doi:10.1145/1462055.1462056.
- [18] X. Bonaventura, M. Feixas, M. Sbert, Viewpoint information, in: *Proceedings of the International Conference GraphiCon'2011*, Moscow (Russia), 2011, pp. 16–19.
- [19] N. Marsaglia, Y. Kawakami, S. D. Schwartz, S. Fields, H. Childs, An entropy-based approach for identifying user-preferred camera positions, in: *2021 IEEE 11th Symposium on Large Data Analysis and Visualization (LDAV)*, IEEE, 2021, pp. 73–83. doi:10.1109/LDAV53230.2021.00015.
- [20] S. Zeng, G. Geng, M. Zhou, Automatic representative view selection of a 3d cultural relic using depth variation entropy and depth distribution entropy, *Entropy* 23 (2021) 1561. doi:10.3390/e23121561.
- [21] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, A. Bousseau, Single-image svbrdf capture with a rendering-aware deep network, *ACM Transactions on Graphics (ToG)* 37 (2018) 1–15. doi:10.1145/3197517.3201378.