



**HAL**  
open science

# Improving the Fidelity of Synthesized Images by Integrating Real-World Data

Reimmer Meldrick, Chatoux Hermine, Olivier Aubreton

► **To cite this version:**

Reimmer Meldrick, Chatoux Hermine, Olivier Aubreton. Improving the Fidelity of Synthesized Images by Integrating Real-World Data. *Traitement et l'Analyse de l'Information : Méthodes et Applications, TAIMA-2023, Arts-pi, May 2023, Hammamet, Tunisia. pp.236 - 249. hal-04189868*

**HAL Id: hal-04189868**

**<https://hal.science/hal-04189868>**

Submitted on 5 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Improving the Fidelity of Synthesized Images by Integrating Real-World Data

Reimmer Meldrick, Chatoux Hermine et Aubreton Olivier.

Université de Bourgogne,  
Laboratoire ImViA,  
9 avenue Alain Savary,  
BP 47870 – 21078 Dijon Cedex, France.  
Tél : +33 (0)3 80 39 38 78,  
[meldrick.reimmer@u-bourgogne.fr](mailto:meldrick.reimmer@u-bourgogne.fr)

**Résumé** Un pipeline pour la synthèse d'images réalistes en imagerie couleur est proposé. L'approche tient compte de la propriété radiométrique cruciale des fonctions de sensibilité spectrale de la caméra, qui joue un rôle important dans la mesure et l'interaction de la lumière avec les objets capturés par la caméra. En incorporant les paramètres géométriques et radiométriques de la caméra dans le logiciel Blender, des images synthétiques sont générées qui sont plus représentatives des scènes du monde réel. Dans une série d'expériences, l'importance de la prise en compte de ces paramètres dans le processus de synthèse est démontrée, et il est démontré que l'approche proposée surpasse les méthodes de pointe qui les négligent.

**Mots clés** Synthèse d'images, Imagerie couleur, Calibration géométrique et radiométrique, Blender.

**Abstract** A pipeline for synthesizing realistic images in color imaging is proposed. The approach takes into account the crucial radiometric property of the camera's spectral sensitivity functions, which plays an important role in the measurement and interaction of light with the objects captured by the camera. By incorporating both the geometric and radiometric parameters of the camera into the Blender software, synthetic images are generated that are more representative of real-world scenes. In a series of experiments, the importance of considering these parameters in the synthesis process is demonstrated, and it is shown that the proposed approach outperforms state-of-the-art methods that neglect them.

**Key words** Image Synthesis, Color imaging, Geometric and Radiometric Calibration, Blender.

## 1 Introduction

Image synthesis refers to the creation of images by artificial means, and these generated images can have a lifelike appearance. This makes them valuable for the evaluation and training of computer vision models, as shown by several studies [1, 2, 3, 4]. However, the effectiveness of these models depends on the availability of large amounts of data, which is not always the case. In these instances, pre-training on synthetic data can be used as a solution. This raises the question of the realism required in the synthetic images in order to effectively train the models.

Most widely used methods for image synthesis is a machine learning model called Generative Adversarial Networks (GANs), which have proven to be an effective solution for image generation in cases where training data is limited. The different GAN variants [5, 6, 7, 8, 9] demonstrate the versatility and power of GANs in solving computer vision problems. GAN-based methods are prone to generating false images, which can be easily detected by spectral analysis. Recent research [10] alleviates this problem.

In addition to GANs, there are other few rendering software's such as Blender [11], Unreal Engine [12], Unity [13] and Rhinoceros 3D [14] for generating realistic images. Blender, an open-source software that permits the creation of scenes and objects and integrates custom parameters using a third-party Python API, was employed in this study. Additionally, BlenderProc [15] is a pipeline built on top of Blender that provides supplementary features such as real images, segmentation masks, and depth. However, the generated image's accuracy can be negatively impacted by inaccuracies in the input image formation, particularly concerning color, which can subsequently affect the precision of computer vision tasks reliant on color information. To tackle this issue, the proposed approach takes a novel perspective and avoids using prior input, which distinguishes it from existing methods.

To enhance the realism of generated images, an approach is proposed that integrates the geometric and radiometric parameters of the actual camera and relevant object information (shape and reflection) into the Blender software [11], as illustrated in **FIG.1**. This integration enables the precise capture of scene attributes, facilitating the creation of high-quality synthetic datasets that accurately represent real-world scenarios. By ensuring the quality of the generated images, the limitations

of current state-of-the-art methods can be overcome, resulting in images suitable for use as training data for computer vision models.

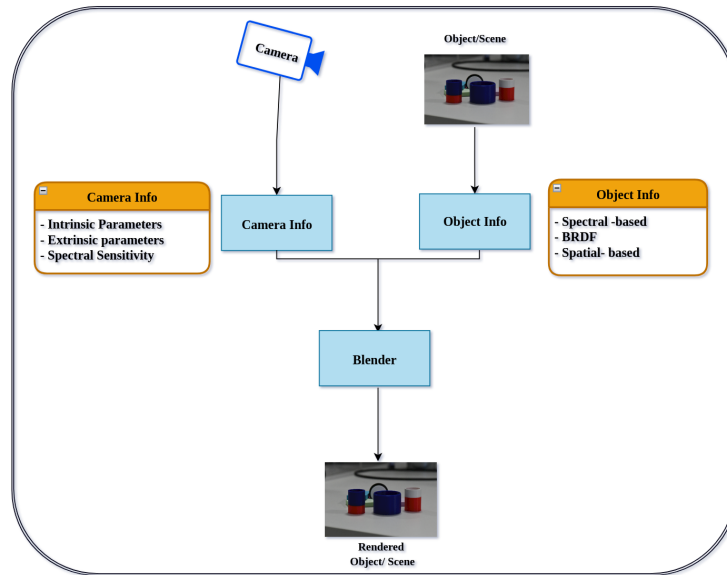


Figure 1 – An illustration of the proposed method for synthesizing RGB images. The orange boxes show the key components that need to be included in Blender for effective rendering. This includes essential information such as the RGB camera’s intrinsic and extrinsic parameters, the camera’s spectral response, the object’s spectral reflectance and Bidirectional reflectance distribution function (BRDF), and spatial features.

The methodology for obtaining these parameters is outlined in Section 2, and the integration of these parameters into the Blender rendering software is described in Section 3. Section 4 presents the evaluation results, including both qualitative and quantitative data, and the conclusion provides insight into the limitations of the approach.

## 2 Experimental Setup

### 2.1 Characteristics of the Camera

The Nikon DSLR<sup>1</sup> D-850 RGB camera was used for the acquisition of the objects and the scene. This is a professional full-frame camera, which has a 45.7Mp BSI CMOS full-frame FX sensor.

The camera has been through a calibration process for the estimation of its geometric parameters. This step is crucial as it provides valuable metric information about the scene and is a prerequisite for subsequent processing steps as seen in **FIG.1**.

The calibration was performed using Zhang’s method [16], which solves a set of non-linear equations relating the 3D coordinates of the calibration pattern corners to their 2D counterparts in the image. The resulting camera matrix is given by the following equation:

$$\begin{bmatrix} 1.34 \times 10^4 & 0 & 1.89 \times 10^3 \\ 0 & 1.35 \times 10^4 & 1.46 \times 10^3 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (1)$$

The first and second rows of the matrix represent the focal length of the camera in the  $(x, y)$  directions, respectively. The third column represents the position of camera principal point in  $(x, y)$  directions. The third element is the ratio of the camera to the real world, which is always set to 1.

In order to ensure the accuracy of the captured scene or object’s image appearance with the Nikon D850 camera, it was imperative to address the radiometric properties, with a specific focus on the camera’s color reproduction. To achieve this goal, a camera characterization was conducted, resulting in the acquisition of the Nikon D850’s spectral sensitivity functions.

Camera color characterization is one of the most important steps in the pipeline to ensure that the camera module is not producing false colors in the processed image. Knowing this characterization allows us to know exactly what to expect and use in subsequent steps.

---

1. Digital Single-Lens Reflex.

The characterization of the Nikon D850 camera has been obtained using the OL750, a spectro-radiometric measurement system, following similar methods done in [17, 18].

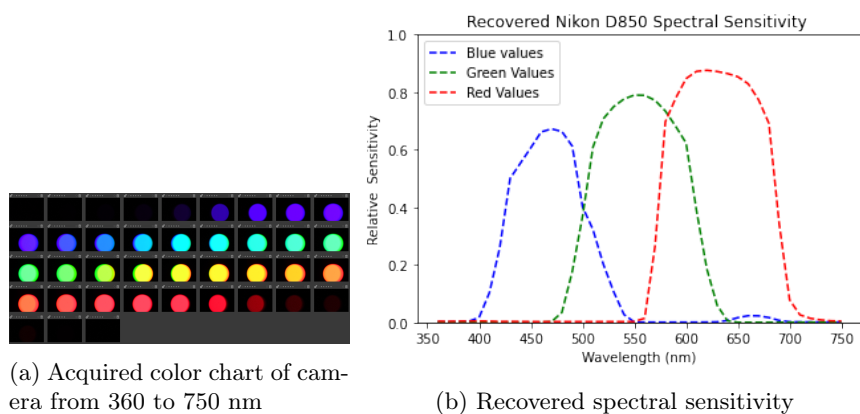


Figure 2 – Characterization of the Nikon D850. From (a) The spectral resolution is 10 nm and the image integration time is 5s using the monochromator.

**FIG.2**, presents a characterization of the spectral sensitivity functions of the Nikon D850 DSLR sensor in color imaging. The blue channel shows lower sensitivity compared to the green and red channels, as observed in human vision. In the blue channel sensitivity curve, there is a small bump in the red channel sensitivity range between 620 and 700 nm. These results demonstrate the distinct spectral sensitivity functions of the Nikon D850 sensor.

## 2.2 Characteristics of the Object

The process of achieving the necessary characteristics of each object to be used in the Blender synthesis is discussed here. Three different objects were modeled with three different colors as shown in **FIG.2B**. The objects were 3D printed and their properties measured.

Spectral reflectance is a metric used to quantify the amount of light reflected by an object as a function of wavelength. This was estimated using a CS-1000 spectrophotometer in a D65 light cabinet. Each object was placed in the light cabinet and its reflectance was measured with the CS-1000.

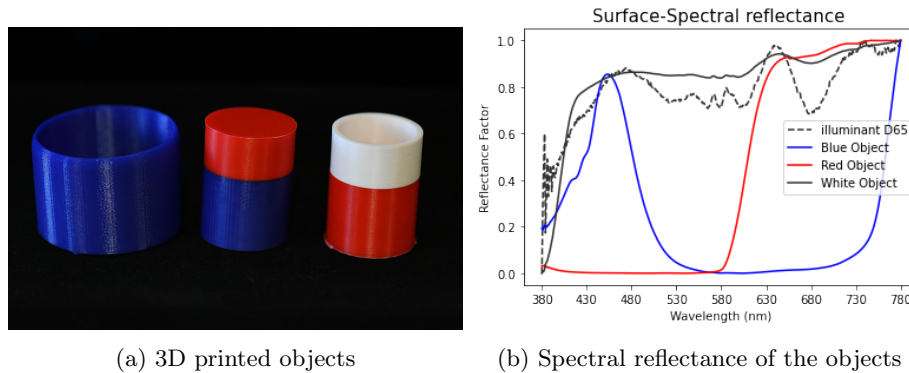


Figure 3 – Objects and their Reflectance. **(a)**, an illustration of the 3D printed objects utilized in this study, and **(b)**, the surface spectral reflectance measurement of the objects.

In **FIG.3**, the spectral characterization of the object is depicted. The red and white regions of the object are clearly characterized by their high reflectance values in the corresponding spectral regions. The recognizable blue region of the object shows a perceived color shift toward red due to reflectance values in the higher wavelength region of 738 nm. This observed shift is contrary to the initial expectations (we perceived the object more blue than purple) and highlights the importance of carefully considering reflectance values when designing synthetic objects for computer vision applications.

### 3 Image Synthesis Procedure

#### 3.1 Blender Software

The Blender software was utilized to create the 3D representation of the scene and objects. The camera’s intrinsic parameters and those of the objects were incorporated into the Blender software to ensure accurate rendering. Blender was used in this study due to its use in research community and it allows third party developments.

However, Blender’s camera does not have a camera sensitivity functions property, so color correction was applied during the pre-compositing process before rendering. This can be done by using compositing nodes in the 3D view-port, where you can apply effects such as color correction, to the 3D scene before it is rendered. The

measured RGB values were used in the compositing pipeline to adjust the color balance and tone of the final output image. The images used in this study are presented below.

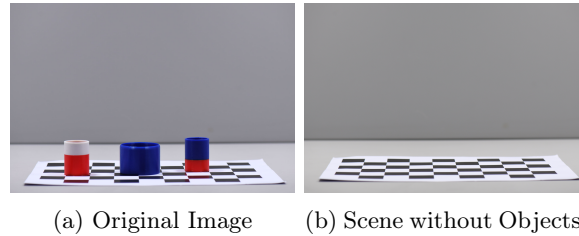


Figure 4 – Images to be used in this study. **(a)**, the original image used as reference in this study, illustrating the scene with objects. **(b)**, Image of the scene without the objects

To use the attained camera matrix in Blender, the focal lengths need to be converted from pixels to millimeters using the pixel size.

Blender takes a single-lens value so, the average of  $x$  and  $y$  focal lengths was computed:

$$f_B = 41.4 \text{ mm.} \quad (2)$$

Also, the  $x_B$  and  $y_B$  coordinate of the principal point would be **0.55** and **0.4** respectively.

The pre-compositing process was carried out using the camera model in Blender, which involved several steps being established.

1. Object Selection: each object is selected for pre-compositing, and a new material is created for the object in the Material Properties tab, giving it a unique name.
2. UV Unwrapping: after obtaining the texture map for the objects, an image-based texture UV mapping [19, 20] is performed to project a 2D texture of the image onto the surface of the object in 3D. This helps to create a texture mapping that is closest to the real image.
3. Color Correction: a color ramp node is added to the object's material, allowing for color adjustment using custom RGB values.



4. Normal Map: a normal map is added to enhance the realism of the object, including surface details such as bumps and ridges.
5. Sampling Lighting: a blackbody node is added to the material to sample the lighting in the scene and adjust the object's colors accordingly, simulating how light interacts with the object's surface.

These pre-compositing steps helped to achieve a more realistic render with accurate colors and surface details that closely matched the real image.

## 4 Experiments and Discussions

### 4.1 Experimental Protocol

The effectiveness of the approach was evaluated using four scenarios, which either included or excluded color and texture processing. In **FIG.4**, represents the original image (4a) used as ground-truth for the comparison and the background of the scene (4b). The steps outlined in Section.3 were used for each scenario.

In "Without texture and SSF," only steps 1 and 5 were applied. In "Without texture, with SSF," steps 1, 3, and 5 were applied. In "With texture, without SSF," steps 1, 2, 4, and 5 were applied. In "With texture and SSF," all five steps were applied.

To comprehensively assess the quality of the images, both subjective and objective evaluations were performed.

### 4.2 Qualitative Evaluation

**FIG.5**, illustrates the effectiveness of the proposed image rendering method. Each image represents the difference between the original and rendered image. The color bar in the figure provides an intuitive visual representation of the mapping between the color values and the corresponding pixel intensities in the original image. A value of 0 on the color bar (white) represents no change in pixel intensity, a red color indicates the higher value in the original image and blue color indicates her intensity in the rendered image.

In the first row, corresponding to the rendered image without color and texture, it can be seen that positioning the camera in Blender to match the original image

## Comparing the pixel profiles of the rendered images (Blue Channel).

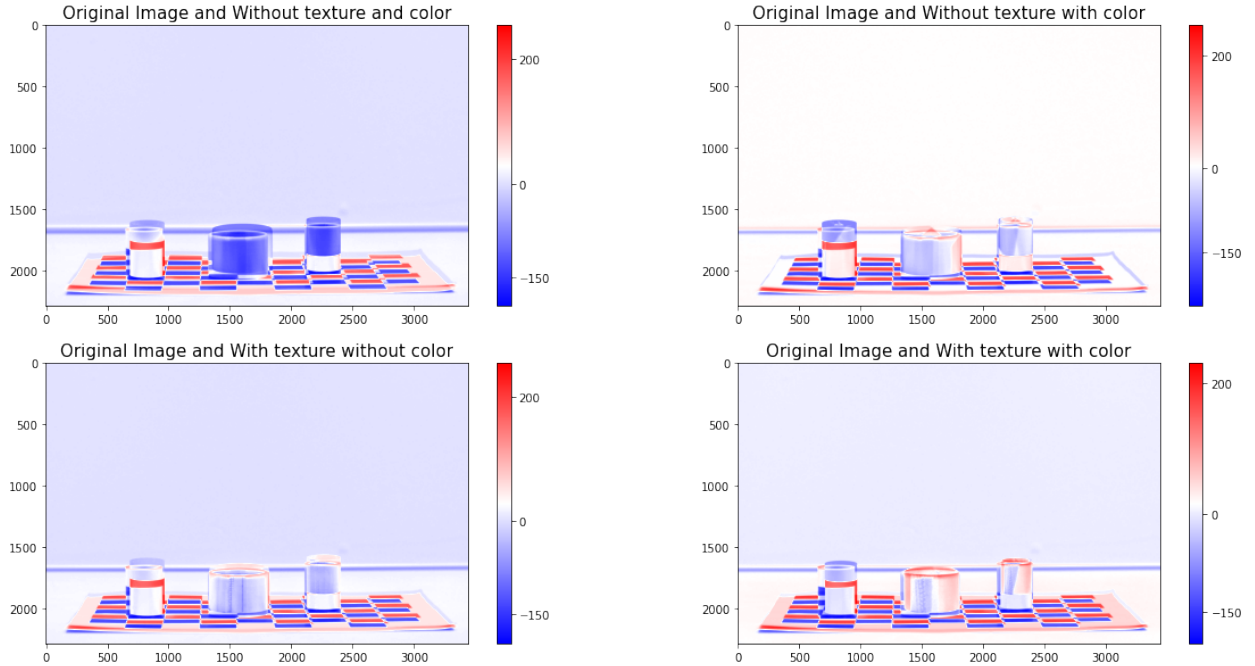


Figure 5 – The graphical representation of the comparison between the pixel intensity of the original and rendered images. The pixel intensity of the blue channel is plotted along the Y-axis, and each column of the image is represented along the X-axis.

can be challenging and result in a minimal error. The central object shows higher differences between the original image and the rendered image without color and texture, highlighting the effectiveness of the proposed method.

In the second row, corresponding to the rendered image with texture, there is a noticeable increase in specular artifacts in the middle and right objects. This is due to the fact that the 2D texture used for projection was an image of a pre-curved object, which exacerbates these artifacts.

Finally, in the second column, which corresponds to the color rendering, the right side of the centered objects shows higher values for the rendered image. This is due to the fact that Blender does not take into account inter-reflections between objects. In the original image, the red object reflects onto the center object, contributing to a more purple color on that side of the object.

### 4.3 Quantitative Evaluation

The quantitative evaluation of the generated images was performed using three metrics: SSIM<sup>2</sup> [21], PSNR<sup>3</sup>, and  $\Delta E$ . The SSIM compares the similarity between the original image and the image generated by Blender, taking into consideration lighting, contrast, and structure. An SSIM value of 1 indicates that the images are identical, while a value of 0 indicates complete dissimilarity. The PSNR is a measure of image quality, with a higher value indicating higher image quality. Finally, the color difference between two images was compared using  $\Delta E$ , which involves transforming the color into the CIELAB color space and computing the difference between the two colors. A smaller  $\Delta E$  value indicates closer color matches.

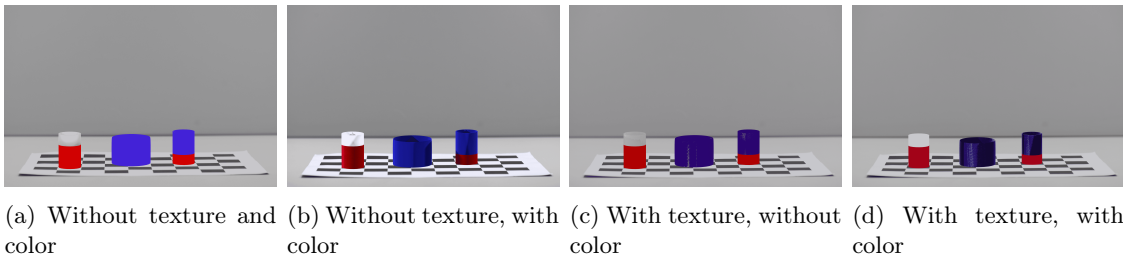


Figure 6 – The results of the approach, as described in **Section.4.1**, are presented in (a) to (d). (a) depicts the rendered image without texture and displays the measured geometric and radiometric metrics. (b) shows the rendered image without texture, but with the measured geometric and radiometric metrics included. (c) presents the rendered image with texture, but without the measured geometric and radiometric metrics. Finally, (d) depicts the rendered image with both texture and the measured geometric and radiometric metrics.

Table 1 – Comparative analysis metrics for assessing the quality of rendered images.

Method	SSIM	PSNR	$\Delta E$
Without texture and SSF	0.91	64.3	10.2
Without texture, with SSF	0.91	65.2	<b>8.3</b>
With texture, without SSF	0.91	65.2	8.6
With texture and SSF	<b>0.92</b>	<b>65.3</b>	8.4

2. Structure Similarity Index Measure.

3. Peak Signal to Noise Ratio.

The results presented in **TAB.1** demonstrate the effectiveness of the proposed method in achieving a balance between texture and color quality in rendered images. This method accounts for the trade-off between texture and color quality, as different methods may be more appropriate for different applications. For instance, the method with the highest SSIM score of 0.92 resulted in a PSNR of 65.3 dB and a color difference of 8.4, as measured by  $\Delta E$ . However, another method with a slightly lower SSIM score of 0.91 had a lower PSNR of 65.2 dB, but a lower color difference of 8.3, indicating that while the application of the texture process improves the SSIM, it may result in a degradation of color quality, as measured by  $\Delta E$ . On the other hand, if the texture process is omitted, the color quality may improve, but the overall appearance may be less visually similar to the original image.

To further improve the proposed method, the integration of a better image-based texture is crucial. A high-quality texture can improve the visual appeal and realism of a rendered image while maintaining a small color difference and increasing the SSIM score. By incorporating a better texture quality, the proposed method can achieve a balance between texture and color quality in rendered images, making it more suitable for applications that require both texture and color to be of high quality. The use of high quality textures is a critical aspect in achieving the desired balance between texture and color quality in rendered images.

## 5 Conclusion

The results of this study show the importance of incorporating real-world camera and object metrics into the Blender rendering process for more realistic images. We designed our proposition to respect the meteorology, the image information are directly linked to the physical properties of the scene. The evaluation of the test scenes using metrics such as SSIM, PSNR and  $\Delta E$  showed an improvement in the realism of the images after incorporating the camera and object parameters.

However, a limitation of this approach was that color correction was only achieved through post-processing compositing and not through the Blender camera itself. Further research in this area should focus on incorporating the color response properties of the camera, thus having a custom color management to improve the accuracy of the rendered images.

## Bibliography

- [1] Hao Su, Charles Ruizhongtai Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: viewpoint estimation in images using cnns trained with rendered 3d model views. *CoRR*, abs/1505.05641, 2015.
- [2] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. 10 2017.
- [3] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015.
- [4] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from RGB images. *CoRR*, abs/1902.01275, 2019.
- [5] Oliver Serang. Adversarial network training using higher-order moments in a modified wasserstein distance, 10 2022.
- [6] Xudong Mao, Qing Li, Haoran Xie, Raymond Lau, Wang Zhen, and Stephen Smolley. Least squares generative adversarial networks. pages 2813–2821, 10 2017.
- [7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 11 2015.
- [8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains, 12 2019.
- [9] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. pages 8798–8807, 06 2018.
- [10] C. Dong, A. Kumar, and E. Liu. Think twice before detecting gan-generated fake images from their spectral domain imprints. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7855–7864, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.
- [11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

- [12] Epic Games. Unreal engine [online]. available: <https://www.unrealengine.com>. 2019.
- [13] John K Haas. A history of the unity game engine. 2014.
- [14] Robert McNeel et al. Rhinoceros 3d, version 6.0. *Robert McNeel & Associates, Seattle, WA*, 2010.
- [15] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023.
- [16] Wilhelm Burger. Zhang’s camera calibration algorithm: In-depth tutorial and implementation, 05 2016.
- [17] Junichi Nakamura. Image sensors and signal processing for digital still cameras. 2005.
- [18] Christian Mauer. Measurement of the spectral response of digital cameras. 2010.
- [19] Guilherme Oliveira, Rafael Torchelsen, João Comba, Marcelo Walter, and Rui Bastos. Geodesic-driven visual effects over complex surfaces. *The Visual Computer*, 27:917–928, 10 2011.
- [20] Robert Jagnow, Julie Dorsey, and Holly Rushmeier. Evaluation of methods for approximating shapes used to synthesize 3d solid textures. *TAP*, 4, 02 2008.
- [21] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020.