3D reconstruction of fairly reflective object using MASt3R algorithm

Final Project Report

Richardson Mondouji, Louise Gabion

***Abstract*—This paper is the final report for a multi-views 3D reconstruction project. Numerical geometry reconstruction is a major topic nowadays for new technologies and historical preservation. We choose to deal with reflective objects of different strength using the MASt3R algorithm and to reconstruct their appearance with Bidirectional Scattering Distribution Function in Blender. We use the PANDORA dataset. We have got effective results for slightly scattering object but we encounter a limitation for strongly scattering materials.**

A step by step tutorial for the method, and extra results are available on the GitHub: https://github.com/Hysa0/3D_Vision_Project.git.

I. Introduction

Reconstructing 3D geometry has still remained an important challenge in many domains, especially for new technology advance and in numerical production. We can cite multiple applications such as autonomous car's environment recognition, video game character's movement, scene reconstruction in extended reality, and even monuments and historical features reproduction for a numerical preservation.

Currently, many solutions have been provided, and different free to use algorithms are available for different applications. However, some parts of this topics are still in research, for more accurate results, less computation time or for less input information's need. In this report, we have choose to mainly focus on the reconstruction of reflective object, where light reflection can be a problem during the matching point phase.

In fact, for reflective materials (even if they are not highly reflective), their appearance depends on the view due to the light reflection on it. Hence, this reflection can be approximate as a noise in the images and create some mismatch during the matching point phase. Because it works like occlusion, it is difficult to precisely reconstruct the actual shape of the object and its glowing aspect. With that in mind, we choose to use the MASt3R algorithm (Matching And Stereo 3D Reconstruction) [1] to reconstruct the object and then use some BRDF reconstruction do give the optical appearance of the original object.

In this report firstly is present the state of the art in term of 3D geometry reconstruction with occluded input and for refractive object. Than, we describe our approach to this problem and finally, we set some results.

II. Related work

A. 3D reconstruction

We can cite multiple way to reconstruct an object in 3D. Some use a point map made with laser to estimate the depth, such as in the LiDAR technics when others used image(s). We distinguished mono-view reconstruction to multiple views (or multi-view). Where mono-view reconstruction algorithms (SIDE for Single Image Depth Estimation) roughly works with deep learning algorithm for a depth estimation, multi-views algorithm usually works by analyzing the disparity between images of the same scene. Computing disparity between two or more images can be time consuming and can have trouble due to multiple environment such as homogeneous object, noises and occlusion. However, it is still more practical than LiDAR in the setup way, and it is still more efficient than SIDE in the accuracy result way.

Nevertheless, as say in Introduction, the reflection of light into reflective objects acts like occlusion, that cause problems for multi-view reconstruction. To avoid the problem, some methods had been proposed. In general, these methods proposed to use polarized images and then, to reconstruct the surface appearance with a BRDF computation. However, these methods need to use polarizers for the images acquisition [2], and/or to deal with an image that represent the lighting ambiance around the object [3]. This can increase the computation time and/or be more constraining for the inquisition step.

In the otherside, algorithm such as MASt3R [1] and DUSt3R [4] deals with occlusion quite well, and can be usefull in this case.

B. Aspect reconstruction

Aspect reconstruction is the way to reconstruct the visual appearance of an object, taking into account all parameters that can affect this object. It goes from, as we say earlier, the shape of the objects, textures, the lights around the object and also point of views of the scene/object. All these parameters represent issues that need to be solved in multi-view reconstruction, but it is still a challenge due to sub-factors such as occlusions, illumination changes, or even non-Lambertian materials. Several strategies have arisen to resolve these issues.

There are methods that combine photometric consistency with learned priors, such as Volumetric Scene Reconstruction via Neural Photometric Consistency [5], which learns about geometry and illumination to reconstruct unseen or partially visible occluded regions with consistent shading.

Another method uses learned reflectance models to capture complex surface. NeRD (Neural Reflectance Decomposition) [6] find a way to separate images into reflectance, normals, and illumination components, allowing to reconstruct a realistic result under varying lighting conditions, which is great for translucent and reflective surfaces. We also find Neural Radiance Fields (NeRFs) and Ref-NeRF [7], that allow complex reconstruction under view-dependent effects such as specularities and reflections. All of this by modeling, in addition, the geometry, the luminance and the appearance in a continuous representation.

In that situation, DUSt3R [4] and MASt3R [1] stand out by learning globally consistent dense correspondences across uncalibrated image sets. Although they do not explicitly model BRDF or lighting, their representation captures structural and appearance information, enabling reconstruction with highly occluded objects or appearance variation.

III. Technical Approach

We will use the MASt3R algorithm to reconstruct the shape of the object. MASt3R is one of the best method nowaday to deal with the occlusion issue, as seen previously. This is why we want to use it to deal with the reflection of light that act like occlusion.

MAST3R is an update of DUSt3R (Dense and Unconstrained Stereo 3D Reconstruction) [4].

In the MAST3R's update, Leroy, Cabon, and Revaud [1] propose to improve the matching capabilities of the pointmap regression. For that, MAST3R adds a dense maching head with a new loss function. This matching head alows to extract local features for each images. These features are plugged into a fast neural network matcher and it creates a Feature-based matching information's that are added to the geometrical matching. This new head matching capability overpass the DUSt3R reconstruction overdependance of noise, and also train, in some way, the DUSt3R algorithm to matching.

Moreover, MASt3R adds to DUSt3R a faster computational matching for Nearest Neighbor evaluation. In that, instead of using all pixel of each images and to compare it with all pixels of all other images, at it is done in DUSt3R, MASt3R use an approach based on subsampling. We will take the example of two images I^1 and I^2. The method samples I^1 of k groups of pixels going from U^0 to U^k, and reciprocally in I^2 with groups going from V^1 to V^k. Then, it does a nearest neighbor search comparing U^0 and V^1 and it does the same with the result and U^1 (see Figure 3: Fast reciprocal matching. in Method of Leroy, Cabon, and Revaud [1]). It iterates until each points converge or until the maximal number of iteration is reached. Between all iterations, if points converge, it will be forget for the next iteration computation.

Finally, one major difference between MASt3R and DUSt3R is that MASt3R uses an metric prediction instead of a normalise one, as it is in DUSt3R. It has its importance while we export your geometrical result for the aspect reconstruction.

For the dataset, we choose to use the PANDORA one [2]. The PANDORA dataset is made of three objects, going from a little reflective to almost strongly specular (see Figure 1).

Figure 1: Examples in the PANDORA dataset. On left the *gnome*, on middle the *ceramic owl* and on the right the *black vase*. These pictures are classified from the less reflective to the most reflective.

Moreover, we choose to reconstruct the material appearance of the object using a Bidirectional Reflectance Distribution Function (BRDF) approach.

BRDF fully describes the reflection properties of a surface. It is defined as the ratio of the polarized spectral radiance (or luminance) reflected in one direction, by an element of surface, to the polarized spectral irradiance (or illuminance) on this surface.

$$f_r(\theta, \phi, \theta', \phi', x, y, \lambda, \rho) = \frac{dL_r(\theta', \phi', x, y, \lambda, \rho)}{dE(\theta, \phi, x, y, \lambda, \rho)} \tag{1}$$

with:

- (θ, ϕ) the angle of incidence of the illuminance (E) of the elementary surface.
- (θ', ϕ') the angle of emmitance of the luminance (L_r) of the elementary surface.
- (x, y) the position of the elementary surface.
- ρ the reflectance (or albedo) of the surface.
- λ the wavelength of the incident light source.

As Equation 1 shows, the BRDF function depends on many parameters, that can be roughly difficult to estimate within few pictures.

Moreover, if the material is quite translucent we should compute the Bi-directional Subsurface Scattering Reflectance Distribution Function (BSSRDF) for a better result, witch take into account the inner reflections inside the material that can "emerged" into the elementary analyzed surface (see Figure 2). Than, BRDF is the sum of all other elementary surfaces inner reflection contribution into the analyzed elementary surface.

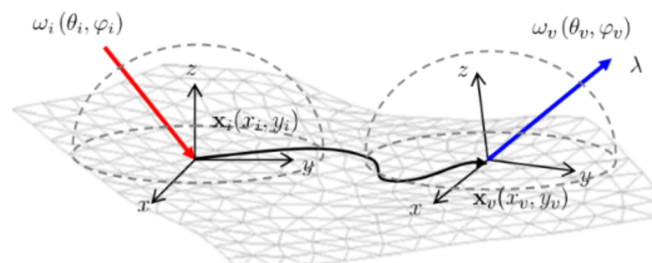

Figure 2: Schematic of the BSSRDF principle by Haindl and Filip [8]

Hence, we preferred to not use translucent material, or to neglect it. For example, for the *ceramic owl*, its glowing aspect is probably due to varnish, witch are translucent, but we prefer to neglect it for now. Hopefully, BRDF computation are wildly used in computer graphics and many computation algorithm are available [3].

However, we also got the possibility to reconstruct the glowing appearance of the object by using a Bidirectional Scattering Distribution Function (BSDF) that mixed BRDF with BTDF (Bidirectional Transmission Distribution Function) [8]. BTDF works like the BRDF but in a transmission measurment. Hence, we should get better results than with BRDF only, but less computing time than with BSSRDF.

IV. Experiments/Analysis

A. Geometrical reconstruction

Using *usage.py* in MASt3R, we can see if effectively it is a good algorithm for slightly reflective object reconstruction. Hopefully, the MASt3R matching points method works well with the *ceramic owl* of the PANDORA dataset [2].

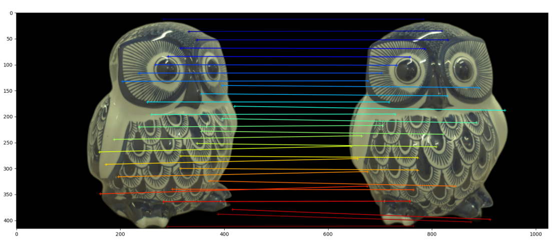

Figure 3: MASt3R 30 matching points between image 01 and 35 of the *ceramic owl*.

However, doing a fast reconstruction using the inside demo of MASt3R, without moving any parameters, the reconstructed object doesn't fit its original appearance, it does not have its reflective aspect anymore (see Figure 4). That confirm the need of an appearance reconstruction method.

Figure 4: On the left: picture of the original *ceramic owl* ; On the right: result of the MASt3R reconstruction

The MASt3R algorithm, even in *demo.py*, comes with multiple parameters including:

- Coarse alignment LR and its number of iteration
- Refinement LR and its number of iteration
- Optimization level between "Coarse" "Refine" and "Refine+Depth"
- Pairing method between "all possible image pair", "sliding windows with our without long range" or to "match one image with all"
- The Truncated Signed Distance Function (TSDF) threshold witch determine the minimum distance of a voxel to be taking into account.
- The virtual camera size witch will be taken to something around 50 cm (so 0.5)

We can also specify if all of our picture shared their intrinsic parameter (in other word if we use the same camera with a fixed focal), witch we firstly assume to be the case for the *ceramic owl*.

The TSDF is fairly correlated to the noise of the result. A low threshold takes more detail into account (voxel close to the surface) but is more noise sensitive. However, a bigger threshold will flattened the reconstruction and we will lose some detailed information. According to the *ceramic owl*, we need to find the good value of TSDF to keep the detail of the leather and the roundness of the owl and in the other hand to not be too sensitive to the noise due to light reflection. However, we have seen that moving the TSDF parameter increase a lot the computing time and the computing resources needed. Indeed, the GTX 1050Ti (Nvidia GPU) wasn't able to run it and always end up with an "Out to Memory" error. But, in the other hand, the TSDF threshold is originally setup to 0 and works like that (witch is still a strange things in our opinion).

In the same way as TSDF, we can also play with the *min_conf_thr* (or the minimal confidence threshold) witch evaluate the quality of a voxel after the TSDF computation. If the voxel quality is lower than the threshold, this voxel is erased. Hence, we have played with it to reduce the noise into our reconstruction.

Figure 5: From left to right: result with *min_conf_thr* = 0.1, result with *min_conf_thr* = 3, result with *min_conf_thr* = 8

Figure 5 shows 3 types of result we have moving only the *min_conf_thr* parameter. In the first case, the *min_conf_thr* parameter is to low, and we have some noise due to the management of "out-of-the-object" point matching. These noises were predictable regarding the matching point result we have in Figure 3 for brown link, around the feet. At the opposite, the last case show a too hight *min_conf_thr*, and where object information's get lost. It shows that the confidence map for the *ceramic owl* reconstruction has lower values for the front of the object than in the back. We also have a similar result with the confidence map in DUSt3R as show in Figure 6.

To conclude for the 3D geometrical reconstruction of the *ceramic owl*, we were able to do it accurately (for quantitative results, see the next section). However, the *ceramic owl* is partially reflective, not strongly specular. What we mean by "strongly specular" is that we can see the reflection of direct light on the surface material, but not the surrounding objects, witch is the case for the black sphere on *black vase*.

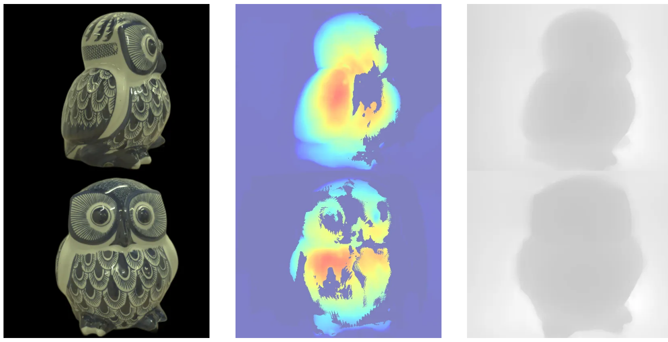

Figure 6: DUSt3R confidence map and depthmap example for *ceramic owl*. From left to right: original image, confidence map and depthmap. Up: image 03.png, Down: image 06.png

In fact, the MASt3R's matching points method doesn't work that well with *black vase*. We can see in Figure 7 that the reflection of the surrounding scene inside the sphere, are considered as an important points by the algorithm.

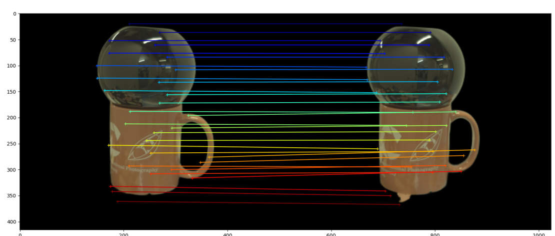

Figure 7: MASt3R 30 matching points between image 01 and 35 of the *black vase*.

As expected, we got a poor reconstruction of the strongly reflective sphere. However, the reconstruction of the cup, with the same appearance of the *ceramic owl*, works well.

Figure 8: Results of the geometrical reconstruction of *black vase* with MASt3R with different values of min_conf_thr

The fact here, is that the DUSt3R's reconstruction works better. We can explain it by the fact that MASt3R match too much points, too well and hence, take into account the surrounding things reflected on the sphere. Nevertheless, this is not the case for DUSt3R, which is not optimized for key points matching [1] [4]. Hence, we obtain a better visually speeking result with DUSt3R (see Figure 9). We insiste in the "visually speeking" because, due to the computation time DUSt3R needs and the lack in our GPU process memory, we can only reconstruct the front side of the *black vase*. Moreover, for the cup, the DUSt3R reconstruction doesn't works well.

Figure 9: DUSt3R reconstruction of *black vase*.

B. Appearance reconstruction

After all the 3D reconstruction result obtain by MASt3R and DUSt3R we worked, as seen in many studies around occluded objects, separately on luminance and occlusion surfaces over the objects we reconstructed. There were many methods that we already quoted in the previous section, but in our study we decided to focus on simulated occlusion and lighting with BRDF and BSSRDF by using algorithms such as Neural geometry and BRDF Reconstruction of Reflective Objects from Multiview Images (NeRO) [3], which seems to be a good match to resolve our problem. But, based on the many compilation issues we had in order to make this algorithm works, we decided to look for another way to separate occlusion rendering by using a 3D model software known as Blender.

B-A. NeRO

NeRO is a recent approach that allows appearance reconstruction by learning a representation of geometry and material properties from mutiview images. This means that unlike traditional rendering pipelines, NeRO does not rely on BRDF and BSSRDF models. However, it works by training a neural network to map position and light direction to radiance values, effectively encoding the object's shape, surface reflectance, and illumination. The strength of NeRO lies in its capacity to learn form image supervision alone and being able to reconstruct objects even under complex lighting and reflectance conditions. NeRO handles implicitly occlusions by learning a volumetric density field that models light absorption, transmission, and scattering. However, since it is a fully neural approach, the meanings of the representation are limited.

B-B. Blender

Blender, on the other hand, allows physically-based rendering (PBR) environment with a full manual control over the simulation of BRDF and BSSRDF, by using the Cycles engine and shader node system. Rather than learning appearance from data, Blender uses explicit shader nodes that simulate effect we can encounter in a real word environment. In Blender, we used 3 different nodes that simulate BRDF and BSSRDF to have our result and compare them to a ground truth dataset based around occlusion. All these 3 nodes work on the principle of BSDF (Bidirectional Scattering Distribution Function), which is a generalization. The BSDF combines BRDF (reflectance principle for surface scattering) and BTDF (transmittance principle for surface scattering); it basically means that if a material does not reflect light only the BRDF is sufficient, but if the object is transparent or translucent we need to compute the BSDF.

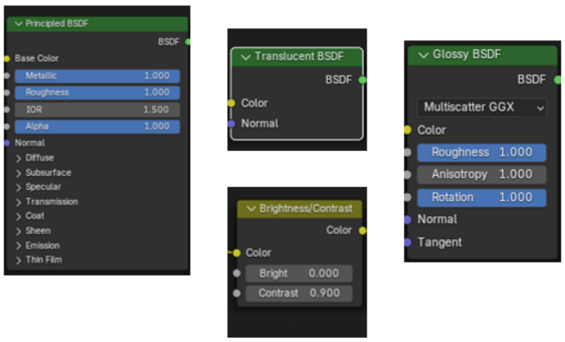

Figure 10: All the principle Node we used to create our BRDF and BSSRDF object, with the goal to mimic the occlusion on object

Now let's present the 3 nodes we used in Blender. First we have the Principled BSDF, this node provide an all-in-one shader based on Disney's model, integrating diffuse, specular, metallic, roughness, and transmission parameters for a flexible BRDF control. There is also the subsurface scattering (SSS) parameters that simulates BSSRDF effects, enabling rendering materials like skin or marble by allowing light to enter the surface and scatter inside.

Secondly, the Translucent BSDF node models is useful to mimic diffuse light transmission through thin materials such as paper or curtains. Physically, it corresponds to subsurface light scattering, where light enters a surface and then exits diffusely without maintaining a preferred direction. The key parameters of the node is Color and Weight (adjustment of the intensity of the translucent effect).

The last one is the Glossy BSDF, which simulates specular reflection, the mirror-like or the rough reflections seen on metals, ceramics and varnished surfaces. The key parameters are the Color (defines the tint of the reflected light), the roughness (controls the sharpness of the reflection 0 for perfect mirror and 1 for completely diffused reflection), the distribution which defines the mathematical model used. For our result we only use the GGX distribution since it is the "most" realistic.

The Translucent BSDF and Glossy BSDF allow us to replicate complex BSSRDF and BRDF behaviors explicitly. Unlike NeRO which estimate implicitly such properties, Blender's node system offers direct and editable control over the visual appearance and light-material interaction.

B-c. Result

Figure 11: Node's application over MASt3R 3D reconstruction of the *ceramic owl*

Recall that all these results come from the same reconstructed file that we showed earlier (see Figure 5). Let us discuss the parameters we have chosen to apply the BRDF and BSSRDF with each node present in Blender. This application enabled us to mimic the occlusion effect that we can observe in the groundtruh image of the *ceramic owl*.

In image A we have the glossy BSDF, the closest node related to the BRDF principle, the setting that was really important to focus on was the roughness. Having a roughness close to zero would change the material scattering to make it reflect light as a metal, which is in our case inconvenient since the owl is in ceramic. The other setting that was important to play with was the brightness and contrast, which is an additional node that we used to adjust the intensity and the contrast of color input. The point here was to find a balance between roughness, brightness and contrast.

In image B, there is the application of the translucent BSDF, the closest node related to the BSSRDF principle, we have here the same scheme, meaning that at first we used the additional bright and contrast node in order to add luminance values and to scale the color around a neutral midpoint. However, in this case the brightness/contrast option was less affecting since the light is diffusely scattered under the material; that is why we did not had to change much this setting because results were already good looking in terms of colors and global luminance.

Despite the fact that in terms of colors and luminance these two nodes were giving acceptable results, visually speaking, we could not see the occlusion as it appears on the groundtruth image of the *ceramic owl*. But finally in image C, we merged the brightness/contrast node with the Principled BSDF node, and the work on the glossy BSDF and Translucent BSDF enabled to create the "occlusion" by finding a balance between roughness, constrast and the new important parameter the metallic setting. It allows us to control whether the surface behaves as a metal or a non-metal surface. This is important because when this parameter is close to 0 we have a fully non-metal surface which reflects white-tinted specular highlights and retains diffuse color, but when it is close to 1 the surface is fully reflective, with no diffuse shading, and specular reflection is colored by the base color. Since we wanted a result as close as the groundtruth image we pushed to 1 this metallic setting and the "occlusion" effect appeared. However, even though the occlusion effect was much better than the other node's application, visually speaking, the result seemed far from the groundtruth image, and this statement led us use metrics in order to evaluate our method.

To do this, we created our own dataset with different views images from our reconstructed images, then we sorted them and applied a algorithm (*Computation.py*) that computes for each image the PSNR (peak signal-to-noise ratio) [6] and the SSIM (structural similarity index) [6]. PSNR (peak signal-to-noise ratio) [9] is a measure used to quantify the quality of an image by comparing the modified (or rendered) image with the reference image. It expresses the difference between the two images in terms of signal-to-noise ratio. In other words, the higher the PSNR, the closer the quality of the modified image is to that of the reference image. Since it is based on the error between pixels of two images, the smaller the difference between pixels, the higher the PSNR.

$$PSNR = 10 \times \log_{10}\left(\frac{(MAX_I)^2}{MSE}\right) \tag{2}$$

- MAX_I is the maximum value for a pixel in an image.
- MSE is the mean square error between the two images.

$$MSE = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_1(i,j) - I_2(i,j))^2 \tag{3}$$

- The formula calculates the average of the square differences between the pixels of the two images, $I_1(i,j)$ and $I_2(i,j)$.
- N and M represent the width and height of the image respectively.

The SSIM [10] is another metric used to measure image quality by comparing two images, it is based on perceptual characteristics such as luminance, texture, and image structure. SSIM measures the similarity between two images by taking three main aspects into account:

- The luminance: the image's light intensity.
- The structure: the way image details are arranged, including patterns and textures.
- The Texture: the fine patterns and structures in the image.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

- μ_x and μ_y: the mean values of images x and y, representing their average brightness.
- σ_x^2 and σ_y^2: the variance of images x and y, which measures the contrast or spread of pixel values.
- σ_{xy}: the covariance between images x and y
- C_1 and C_2: constants (typically, $C_1 = 10^{-8}$ and $C_2 = 10^{-8}$).

With these two metrics we will know on each image where the reconstruction has been performed well and have a valid method to compare images, and thus our method, on the one hand the pixel error and, on the other hand, a more perceptual approach, similar to how we, humans, perceive images.

Dataset	PSNR (Mean Values)	SSIM (Mean Values)
Translucent Owl	13.97 dB	0.6558
Principled Owl	12.76 dB	0.63
Translucent Gnome	14.15 dB	0.79
Principled Gnome	14.09 dB	0.779

Table I: Comparison of PSNR and SSIM Mean Values over 7 images of different view-point of the object for Different Datasets

Since we had to manually create our own dataset, we had chosen to take only seven images of each object to make a mean value calculation over PSNR and SSIM calculation. What is striking is that our PSNR results are not very good, it may come from the voxels (little cube artifact that you can see in our result) that appear after the 3D reconstruction, the PSNR calculation takes account of every pixel value present in the image, so a little variation in the shape or texture of the image can lead to a drastic drop in result. The other parameters that can disturb the calculation is the lighting. The light of Blender is also manually managed so it is very hard to recreate the light that was present in groundtruth images. It was also a problem with the NERF and RENERF method [6], so to partially reduce the issues arising from the lighting we created a script (*Bsdf_code.py*) allowing us to have the same light at the same position and with the same intensity for our study. Another questionable result is that the Translucent BSDF method seems, in both datasets, to have the best result with both metrics. It is probably, because the diffuse color scattering obtained by the translucent method seems to be much more accurate in terms of colors(since there is no occlusion there is no potential color intensity variation), shape and texture (because the diffuse subsurface scattering is masking the voxels). Now, let us check which image has the best score, to see if our assumption is valid no matter the case.

Figure 12: Results based on witch the metrics PNSR and SSIM are performed. From left to right: translucent BSDF, principled BSDF, original picture which was not used for the geometrical reconstruction as input

Visually speaking, when we see all these images we would think that the better results is the translucent one, because the colors match perfectly and the only issue is the quality, the lighting and the occlusion, which is not distinguishable. And the results have:

- for translucent BSDF : PSNR = 15.66 dB and SSIM = 0.70
- for principled BSDF : PSNR = 14.33 dB and SSIM = 0.66

In this case the assumption we made before is valid both visually speaking and also with metrics we used, since the Translucent BSDF gives higher results.

Figure 13: Secondary results based on witch the metrics PNSR and SSIM are performed. From left to right: translucent BSDF, principled BSDF, original picture

Now let us analyse the result for the *gnome* :

- for translucent BSDF : PSNR = 15.82 dB and SSIM = 0.816
- for principled BSDF : PSNR = 16.17 dB and SSIM = 0.817

Here we have a really close SSIM but a better PSNR for the principled BSDF, the reason seems to be the effect of the lighting since in the principled BSDF image we have, visually speaking, highest intensity pixels color or at least the color pixel intensity seems to be closer from the original image; however, the SSIM is the same, so that is why we can say there some bias regarding some result, due to lighting, shape and texture, quality of imported object on blender software and also the node's settings (including additional shader).

V. Conclusion

From this project, we have seen that it is possible to 3D reconstruct slightly reflective objects, easily, using the MASt3R algorithm and to reconstruct the appearance with BSDF models. However, this method is not accurate for strongly scattering materials such as for the *black vase* for which the matching capability of MASt3R is not optimized.

References

[1] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r", in *Computer Vision – ECCV 2024*. Springer Nature Switzerland, Nov. 2024, pp. 71–91, ISBN: 9783031732201. DOI: 10.1007/978-3-031-73220-1_5. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-73220-1_5.

[2] A. Dave, Y. Zhao, and A. Veeraraghavan, "Pandora: Polarization-aided neural decomposition of radiance", in *Computer Vision – ECCV 2022*. Springer Nature Switzerland, 2022, pp. 538–556, ISBN: 9783031200717. DOI: 10.1007/978-3-031-20071-7_32. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-20071-7_32.

[3] Y. Liu, P. Wang, C. Lin, *et al.*, "Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images", *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–22, Jul. 2023, ISSN: 1557-7368. DOI: 10.1145/3592134. [Online]. Available: http://dx.doi.org/10.1145/3592134.

[4] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[5] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections", arXiv, 2020. DOI: 10.48550/ARXIV.2008.02268. [Online]. Available: https://arxiv.org/abs/2008.02268.

[6] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields", in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 5481–5490. DOI: 10.1109/cvpr52688.2022.00541. [Online]. Available: http://dx.doi.org/10.1109/CVPR52688.2022.00541.

[7] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. A. Lensch, "Nerd: Neural reflectance decomposition from image collections", arXiv, 2020. DOI: 10.48550/ARXIV.2012.03918. [Online]. Available: https://arxiv.org/abs/2012.03918.

[8] M. Haindl and J. Filip, *Visual texture: Accurate material appearance measurement, representation and modeling*. Springer Science & Business Media, 2013.

[9] Wikipedia contributors, *Peak signal-to-noise ratio*, Accessed: 2023-05-27, 2023. [Online]. Available: https://fr.wikipedia.org/wiki/Peak_Signal_to_Noise_Ratio.

[10] Wikipedia contributors, *Structural similarity*, Accessed: 2023-05-27, 2023. [Online]. Available: https://fr.wikipedia.org/wiki/Structural_Similarity.