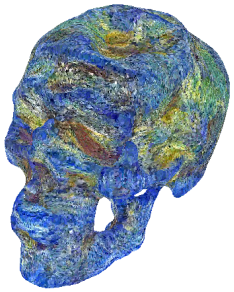
Fusion of 3D Reconstruction and Style Rendering

Fabian Visser

Supervisor 1: Nail Ibrahimli
Supervisor 2: Liangliang Nan

January 16, 2023



1 Introduction

Arguably, two of the most interesting fields in machine learning right now is the development of differentiable 3d reconstruction, and stylistic rendering. Implicit Differentiable Rendering (IDR) recreates 3d geometry, camera parameters, and color and reflectance of the object, requiring only masked images as input. The geometry is modeled using the zero level set of a NN, and the material using a neural renderer.

Stylistic rendering is all about applying styling, artistic characteristics, to a scene. Style representation correlates features between different layers of a CNN, which can be used to reconstruct styling while keeping content representation from another input image.

The task at hand is to find a way to combine these two techniques, using machine learning to create a renderer able to create geometry and styling from a collection of masked images. The progress currently achieved is that using the output pointcloud of IDR, we were able to project it into a 2d image, which has styling applied to it. The styled image is used as

Figure 2: Old post office lit up during the Ghent Light Festival 2011. Applepoi (2011)

pointcloud colouring based on 3d-2d point correspondence, giving an output of a stylized pointcloud.

An application of this can be for quick visualisation of light shows. During the Ghent Light Festival 2011, the old post office was lit up with a multitude of colours using projection (2). To create a quick example of what such an effect would look like on any collection of buildings, or even objects, a user could create their own collection of masked images and include a style image, which are then used to create a 3d geometry with the specified styling.

Figure 3: We can see what van Gogh's style will look like on a section of building.

2 Related work

Implicit Differentiable Renderer (IDR)

Implicit Differentiable Renderer (IDR) developed by Yariv et al. (2020) allows for the identifying of 3D geometry as a zero level-set of an MLP, as well as the lighting conditions and camera parameters of a scene using an input of masked 2D images. The appearance is created by a neural renderer approximated the bidirectional reflectance distribution function of the surface. An interesting coincidence to this approach is the ability to separate the two renderers, applying the appearance from one surface to the geometry of another. IDR can therefore be used to learn geometry, while the appearance can be introduced from elsewhere.

Neural Style Transfer

Gatys et al. (2015) developed a method using generic feature representation to learn and transfer content and style from two different images. A noise image is created and gradient descent is performed on itself and it's Gram matrices entries to match the features of the content image and the Gram matrices entries of the style image respectively. The differences are minimized to create the style transfer image, with weighting factors to adjust content and style representation.

Artistic Render Fields (ARF)

From the novel view renderings created using neural radiance fields, Zhang et al. (2022) develop a method to create a new view-consistent, stylized rendering. VGG feature maps from the rendered and styled image are calculated, and a nearest neighbour matching is between each feature vector at every pixel location of the feature maps. This is chosen over previously mentioned Gram Matrices due to not relying on global aspects, allowing for local style changes to be more apparent.

t-distributed Stochastic Neighbor Embedding

To be able to apply stylization to a 3-dimensional object, or even a 6-dimensional if we include normal information, we look to t-SNE by van der Maaten and Hinton (2008). It allows for the creation of 2-dimensional mapping of higher dimensional data with the ability to tune the perplexity, the importance of local and global aspects to the data. Similarity between datapoints, in the higher dimension as wel as lower one, is calculated based on the probability they neighbour eachother in proportion to their probabilty density based on a Student t-distribution. The mismatch between these two similarities is minimized using a cost function to create a best fitting mapping.

3 Research questions

The research question is as follows,

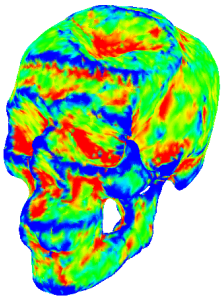
To what extent can a styled 3d reconstruction from a set of content images and a style image be created such that style consistency is present across all views.

The goal is to train a network which is able to reconstruct 3d geometry based on masked images, as well as style the geometry based off of a style image. The 3d reconstruction method as well as the styling will be adapted from previous works, which will be combined into one pipeline. The focus will lie on implementing IDR and NST, looking at combining these two using some kind of 2d mapping, be it tSNE or something simpler such as stereographic projection.

4 Methodology

To start, coloured pointcloud data from IDR will be used as input. Later IDR will be included in the complete process.

The point data will be projected onto a MxN grid and stored as an image. The image will then be used as the content image for style transfer, the new coloring will be applied to the corresponding 3d points. What method for projection will be explored.



(a) Input created using IDR and given arbitrary colors for exemplary purposes

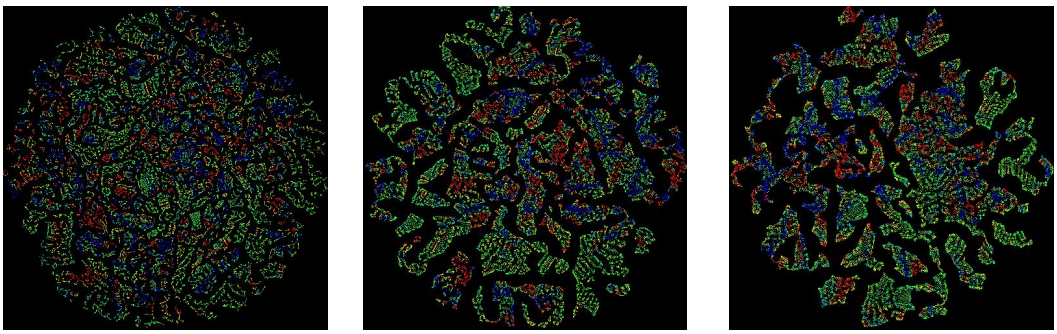
(b) Styling we will apply: van Gogh's Starry Night

4.1 t-SNE

t-SNE can be implemented using the python package scikit-learn. As input we give the pointcloud stored as a numpy array, which is normalized,

$$\frac{P-\mu}{\sigma}$$

Also explored will be if normal information can be included as extra dimensions, to improve style quality. The parameters are set to default, except for the perplexity in the following examples being set to 50. The perplexity can be adjusted based on how much weight should be given to local and global aspects, shown in figure 5. The downside with using t-SNE is it's complexity $O(2N \log N)$, which for large pointclouds can be very slow. The output can be used to teach a MLP to create a mapping faster.



(a) p=5 (b) p=15 (c) p=30

Figure 5: Differences in perplexity. Local data is grouped together more with a higher perplexity, leading to a more localized styling

4.2 Equirectangular projection

A simpler method will also be explored. The pointcloud can be projected onto the unit sphere localized at it's centroid,

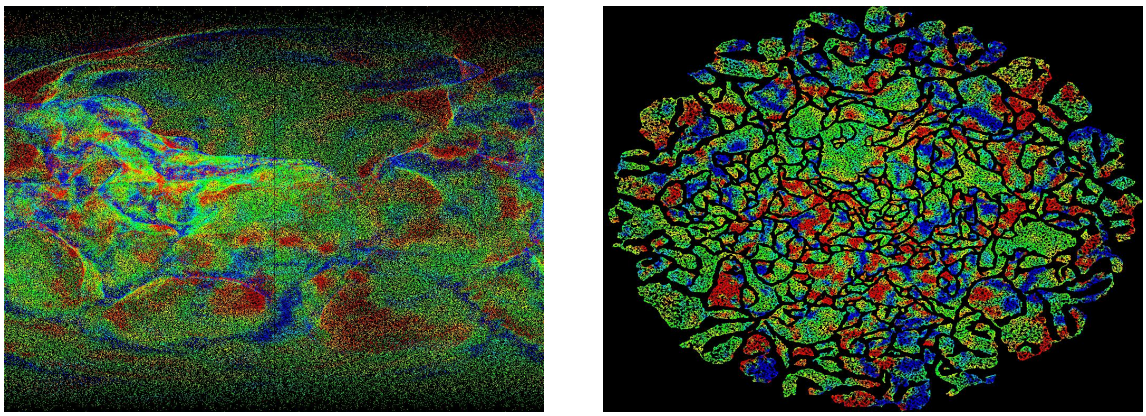
$$\theta = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

$$\phi = \arctan x$$

which is then unwrapped using for example equirectangular projection.

$$u = \frac{M\phi}{2\pi}, v = \frac{N\theta}{\pi}$$

This method is much faster $O(N)$, but has a more defined discontinuity where it is unwrapped.

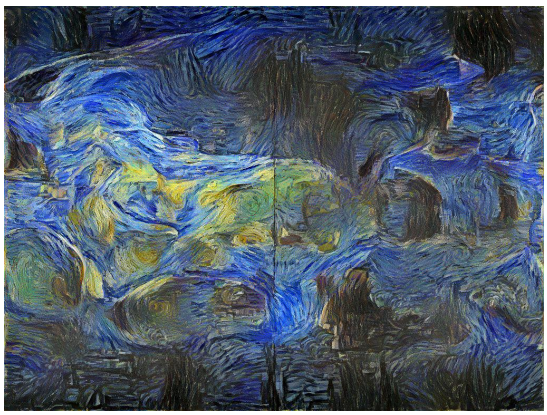


(a) Equirectangular projection (b) t-SNE

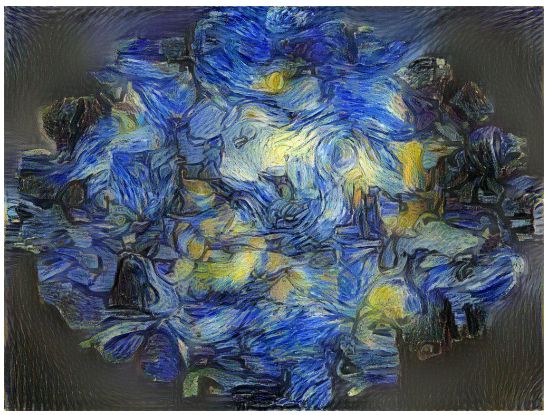
4.3 Styling

Once the mapping is obtained, we can export it as an image, with the pixels' color coming from the corresponding pointcloud's coloring.

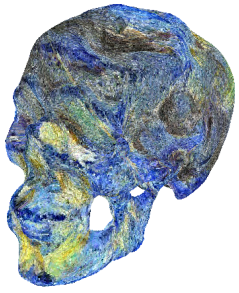
This image is taken as input, with a style image, for the Neural Style Transfer. As output we receive the stylized image of the mapping. The new coloring can then be applied to the corresponding 3d points, giving us a stylised pointcloud.



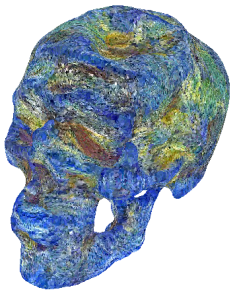
(a) Equirectangular projection



(b) t-SNE



(a) Equirectangular projection



(b) t-SNE

Figure 8: Output pointclouds after reprojecting

We see in the final output that there is some styling consistency. The originally red colours are now represented by grey, and where green was most visible yellow was applied. The actual swirls are however not entirely consistent yet and the resolution of the pointcloud is still rather low.

5 Further work

The next step is to include IDR in the complete pipeline. A style renderer could be placed parallel to or after the neural renderer such that while the geometry and colouring are learned, the styling can be worked on as well. This will also hopefully improve the issue with resolution, as right now a pointcloud has to be created with a stored number of points, while IDR works with implicit geometry allowing for a higher resolution.

Since projecting to 2D will always cause discontinuities (9), we have to look for an approach to optimize for continuous textures, by including it as a loss to minimize or try to create seamless textures.

Also to be explored is the introduction of normal data. Points that lie on a similar plane should have similar styling, while large changes in normal information could lead to large change in style.

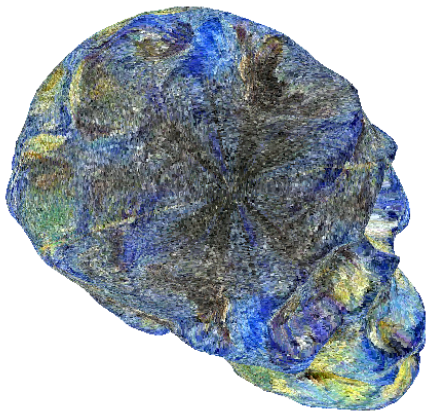
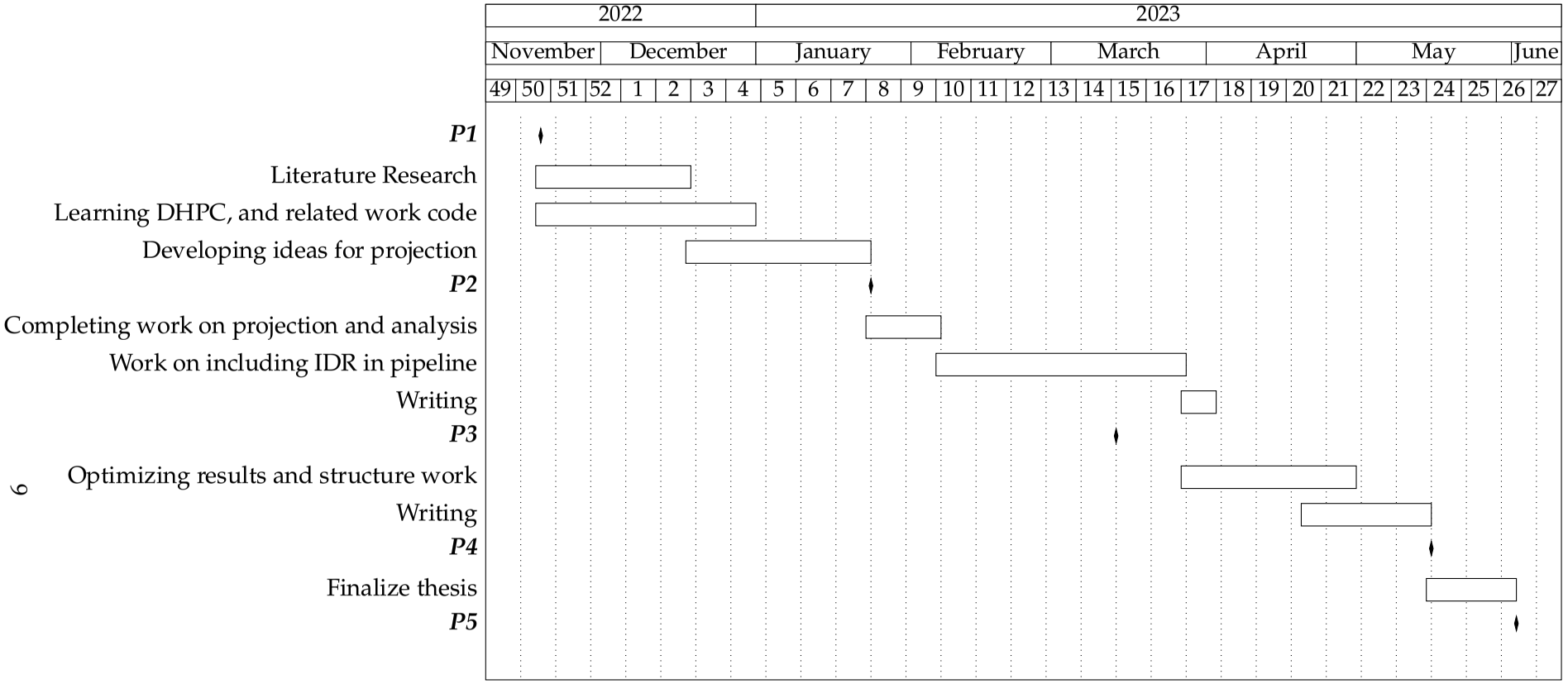


Figure 9: Due to equirectangular projection, we see the styling becomes squished at the poles

6 Time planning



7 Tools and datasets used

The code is written in python, using pytorch for machine learning. The pointclouds and images are manipulated using open3d and pillow respectively. The pointclouds are then viewed using easy3d. The programs are run using the DelftBlue supercomputer.

The datasets are taken from DTU MVS Data Set 2014 by Jensen et al. (2014).

References

Applepoi. Light festival ghent - two, Jan 2011. URL https://www.deviantart.com/applepoi/art/Light-Festival-Ghent-Two-195316421.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style, 2015. URL https://arxiv.org/abs/1508.06576.

R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, R. Basri, and Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance, 2020. URL https://arxiv.org/abs/2003.09852.

K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely. Arf: Artistic radiance fields, 2022. URL https://arxiv.org/abs/2206.06360.